Quantum Computing & Applications for Engineering

10/15/2024

Fault Tolerance & Applications

Announcements

- HW4 is up.
  - Covers up through today.
  - Project 2 groups and proposals.
- Course midterm survey is open.
  - Too hard/easy? Too fast/slow?
  - Balance of hardware, software, math, physics, and applications?
  - Should we have a textbook next time?
  - Should there be an “advanced topics” course?
- IBM Quantum Internships

Today’s goals

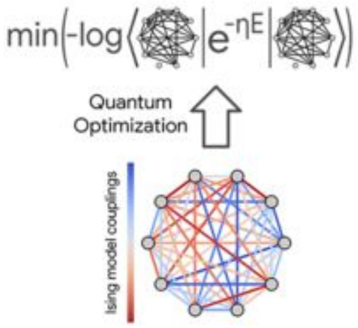
- Paths towards fault tolerance
  - Physical vs. logical qubits
  - Qubit improvements
  - Error correction
- Application Overview
- HW3 Discussion

The Story So Far

- Quantum technologies are based on the control and readout of quantum states (qubits).
  - We apply gates to a statevector to perform computational operations.
  - In real systems this translates to shooting light or radio signals at quantum objects.
- Real systems suffer from noise and other complexities that require careful accounting.
  - We use the density matrix to account for where the information is going.
  - We use extra operators (Kraus operators) to model noise.
- We need to do benchmarking and error control to understand and make use of today's hardware.
  - Single qubit characterization - T_1/T_2
  - Multi-qubit fidelity - Randomized benchmarking, EPLG
  - Application-oriented benchmarks.
  - Error suppression and mitigation.

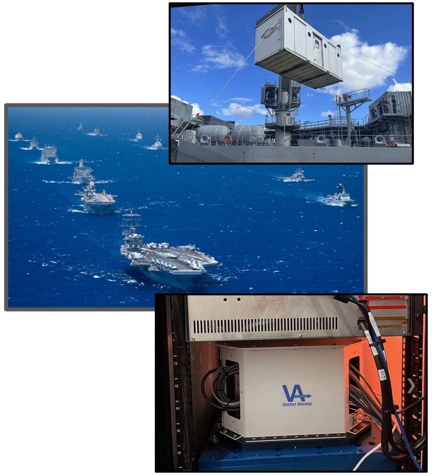
What do we want to do?

- Computing
  - Model and simulate complex materials.
  - Optimize schedules, routes, resources.
  - ~~Break RSA encryption.~~ (We don't actually want to do this)
- Sensing
  - Find our way without GPS
  - Map electrical activity in the brain
  - Verify semiconductor circuits
  - Detect single atoms of substances
- Networking
  - Link quantum computers and sensors together
  - Securely communicate with quantum information

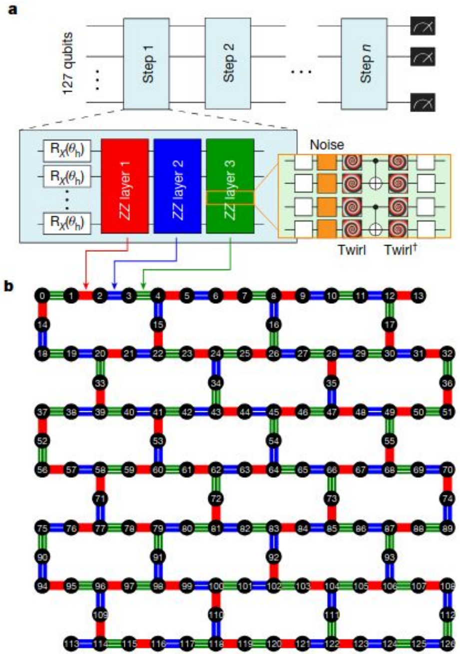


What can we do today?

- Computing
  - Proofs of concept of algorithms and subroutines
  - Physics and physics-like experiments
- Sensing
  - Navigation & Timing
  - Material and device characterizations
  - Biomedical applications
- Networking
  - Simple key-distribution protocols
  - Entanglement exchange and teleportation



At-sea testing of quantum gravitational navigation (Vector Atomic, 2022)



Evidence for the utility of quantum computing before fault tolerance (Nature, 2021)

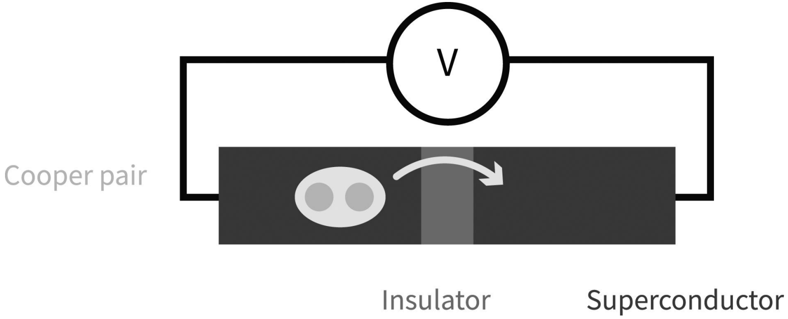
How do we get there?

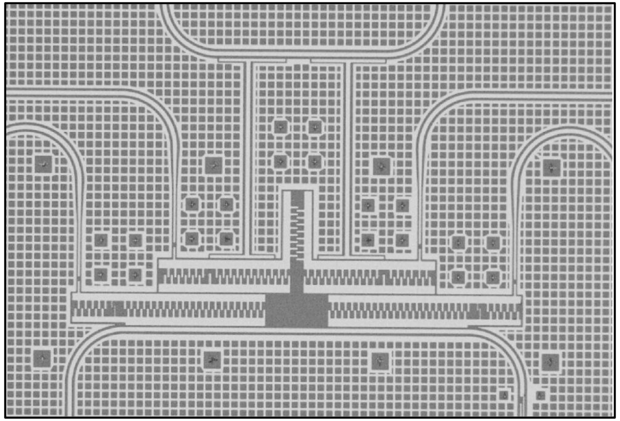
- Make better qubits and controls.
    - Increase coherence times.
    - Improve control hardware and protocols.
- Develop better ways to correct errors using extra qubits.
    - Error correction preserves information even if logical qubits are disrupted.
- Develop ways to convert between qubits.
- Use different physics.
    - Majorana Fermions
    - Topological braiding
- **Fault Tolerance**
    - Do anything with a quantum system as if there were no errors.

Improvements in Superconductors

- Most superconducting qubits are based on the *transmon* architecture.
    - Two superconductors separated by an insulator (Josephson Junction).
    - Coupled to a capacitor to smooth out charge sensitivity.
- Improving the engineering of the components leads to better performance.
    - Geometry & placement
    - Materials
    - Manufacturing precision & consistency

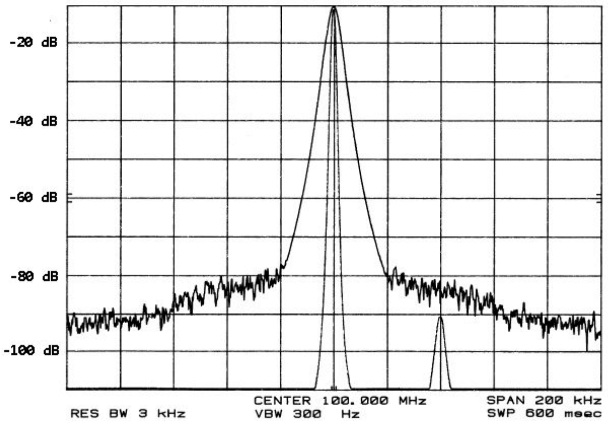
Josephson Junction

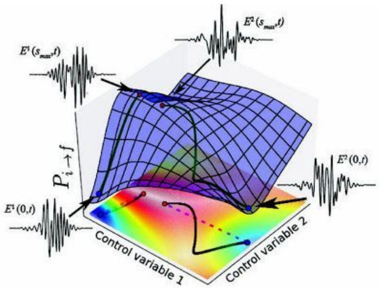


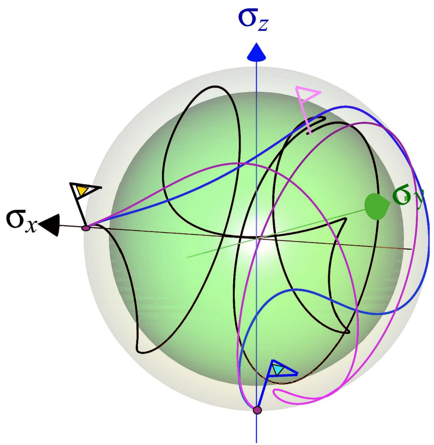


Improvements in Controllers

- Lasers and RF Signal generators are all susceptible to *phase noise*.
  - Small jitters in the frequency
  - Limits how accurately you can apply gates.
- The control pulses themselves can be optimized.
  - Some paths are faster, others are more error-resistant.
  - Controllers are being coupled with reinforcement learning.

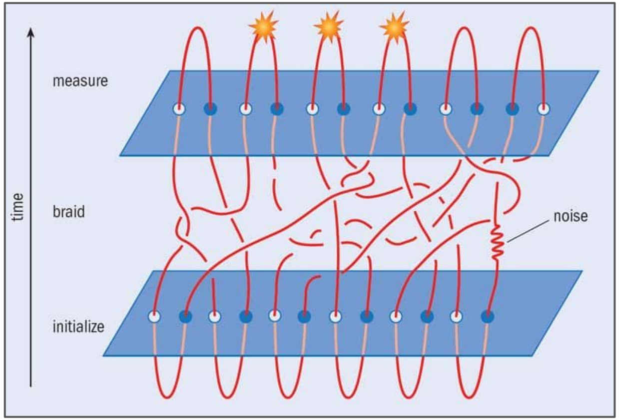




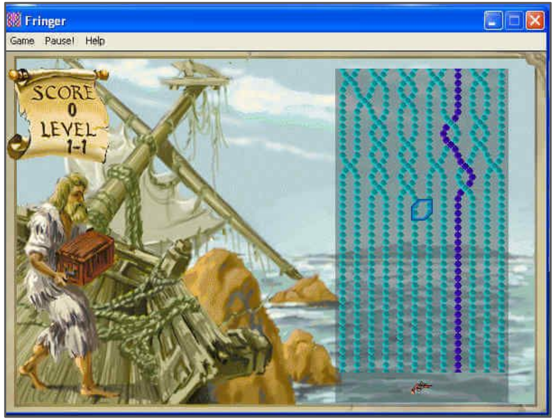


Topological Quantum Computing

- Certain quantum particles form *braids* in space-time.
    - Majorana Fermions
    - Nonabelian Anyons
- Braids are hard to untangle.
    - One you apply a gate, the system gets "stuck" that way.
    - Only the inverse gate will get it unstuck.
- So far nobody has successfully made a topological qubit.
    - Many claims & retractions.
    - Microsoft was pursuing this approach.
- If someone does make one, it could be "the winner."
    - Errors expected below 10^{-6}
    - Logical qubits would be straightforward.



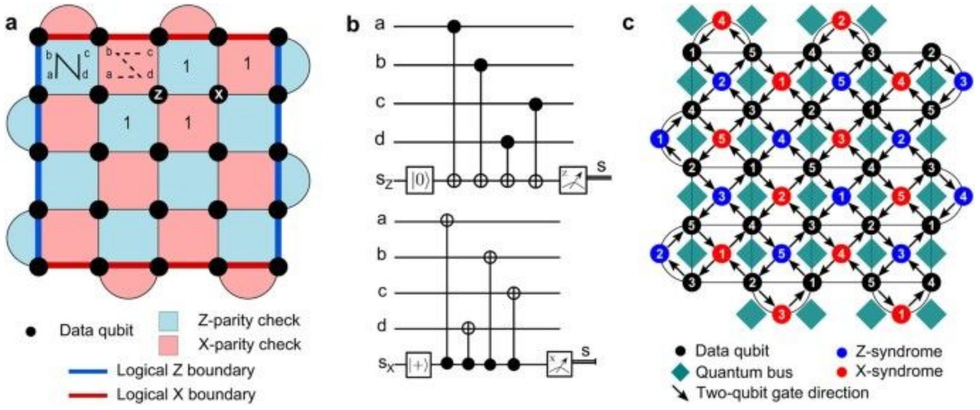
Braiding of qubits (Physics World)



The 1997 game Fringer was Microsoft's first application of braid theory.

Physical vs. Logical Qubits

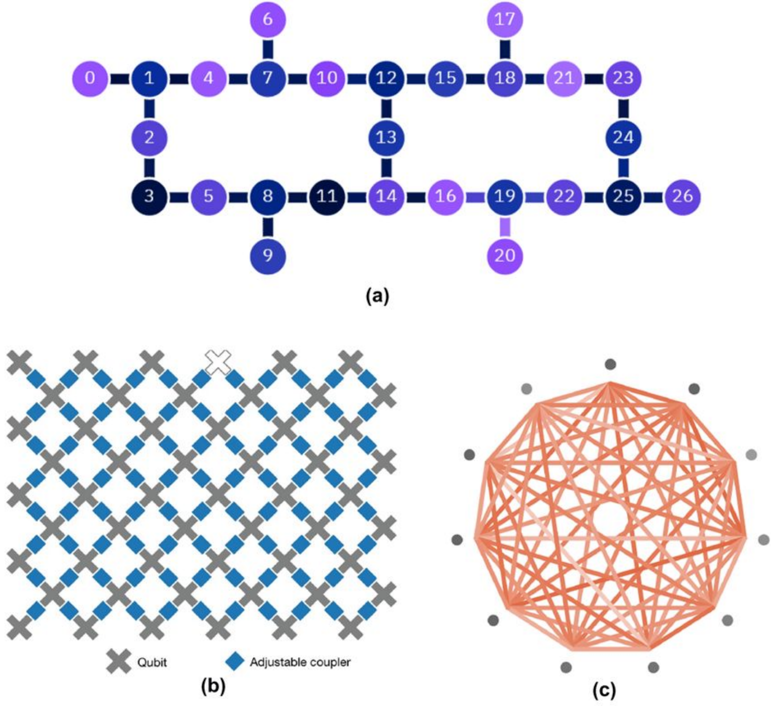
- We are working with physical qubits.
  - Individual elements.
  - Influenced by external noise and imprecise controls.
- Our theories are based on logical qubits.
  - Perfect fidelity, perfect control.
  - Density matrix has no leakage to the classical channels.
- How do we reconcile the two?



Logical Qubits by Google

Encoding and Error Correction

- We can use *redundant* physical qubits to encode information about the state.
- If noise affects a single physical qubit, the others can detect the error and restore the state.
- This encoding enables us to create a true logical qubit.
- Extra qubits are called *ancilla* qubits.



Comments on Error-Correcting Codes

- The entire subject of quantum error correction would take up a graduate-level course.
- Better error-correction codes that reduce the physical overhead are an important research topic.
  - Even if qubits had infinite coherence times, they will still have gate and SPAM errors.
  - Robust error correction codes with the lowest overhead possible are needed.

The Threshold Theorem

- The extra qubits we're using to do error correction are themselves error-prone!
    - Can we even get ahead?
- If we can make qubits with a total error rate below a certain threshold, we can <u>provably</u> correct errors.
- If we can provably correct errors, then we can make logical qubits.
    - The challenge is in the engineering.
    - It can take a lot of physical qubit to make one logical qubit.

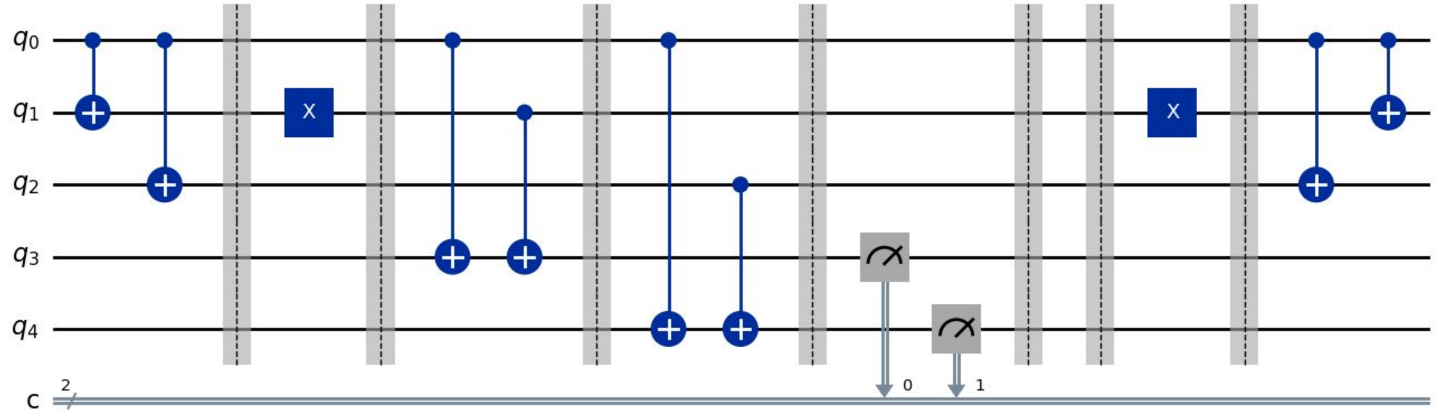
code (QB=qubit)	# physical data qubits	# stabilizers (weight)	typical error threshold range	arbitrary Pauli error?
3-QB bit-flip	3	2 (2)	N/A	no
3-QB phase-flip	3	2 (2)	N/A	no
5-QB Laflamme	5	4 (4)	N/A	yes
7-QB Steane	7	6 (4)	10^{-4} to 10^{-6}	yes
9-QB Shor	9	6 (2) & 2 (6)	10^{-4} to 10^{-6}	yes
17-QB Surface	17	4 (4) & 4 (2)	10^{-2} to 10^{-3}	yes

Basic comparison of quantum error correction codes presented in this course. The stabilizers and weights provide an estimate on the number of ancilla qubits required to achieve fault-tolerance. The estimated error threshold is an indicative range, and depends on the architecture used. Thresholds are not provided (N/A) for the first three codes.

The Repetition Code

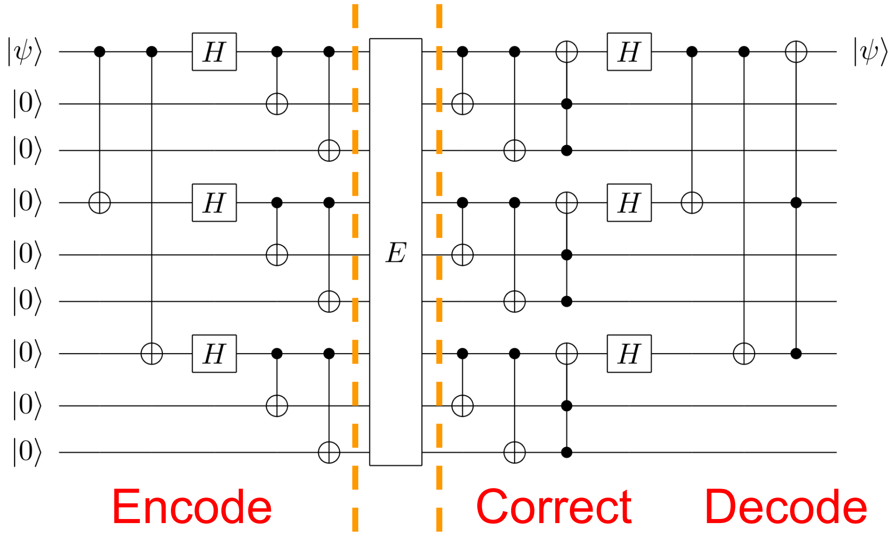
$$|\psi_L\rangle = \alpha|000\rangle + \beta|111\rangle$$

- A naive way to encode a logical qubit is simply to have duplicates.
- We implement a parity check to determine if any bits were flipped.
- If we detect a bit flip, we apply a corrective operation to flip the bit back to its original state.
- Can only correct flips in the X direction. Qubits have an extra degree of freedom!



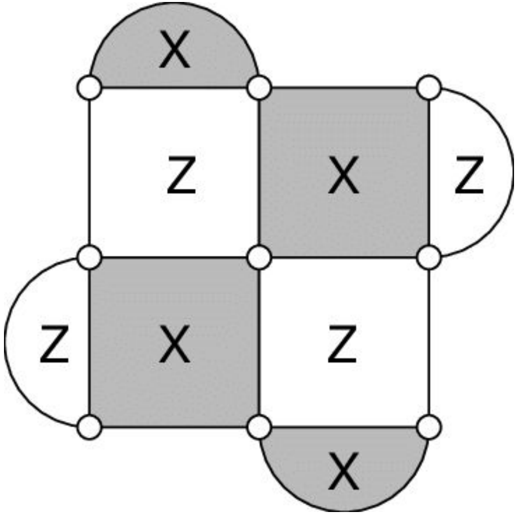
The Shor Codes

- Peter Shor developed 5 and 9 qubit error-correcting codes that can correct bit flips and phase flips.
- These are the mathematical minimum size needed for error correction.
- More ancillas are needed in practice because the qubits themselves have errors.



Surface Codes

- Surface codes use a 2D lattice of qubits to encode logical qubits
  - Good fit for superconductors and other limited-connection architectures
- Can tolerate high single-qubit errors
  - 0.1-1% error is ok
- Potentially large quantum and classical overhead.
  - 100s to 1000s of physical qubits needed per logical qubit.
- This is what most superconducting vendors are looking at.

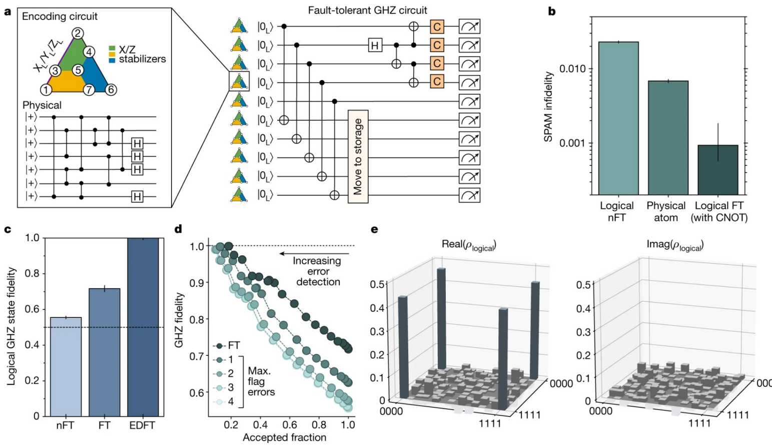


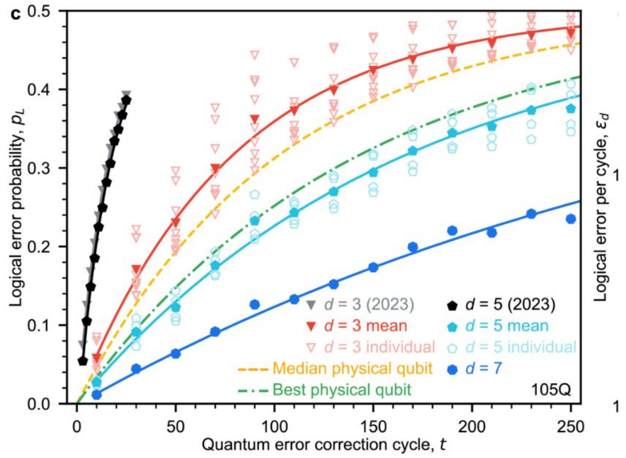
Recent Experiments

- In 2023, Quera and Harvard demonstrated 48 logical qubits in a 256 physical qubit neutral atom lattice. (https://www.nature.com/articles/s41586-023-06927-3)

- In April 2024, Quantinuum and Microsoft demonstrated 4 logical qubits in 30 trapped ions. (https://arxiv.org/pdf/2404.02280)

- In August 2024, Google demonstrated a single logical qubit in 101 superconducting qubits. (https://arxiv.org/pdf/2408.13687v1)





Fault-Tolerant Algorithms

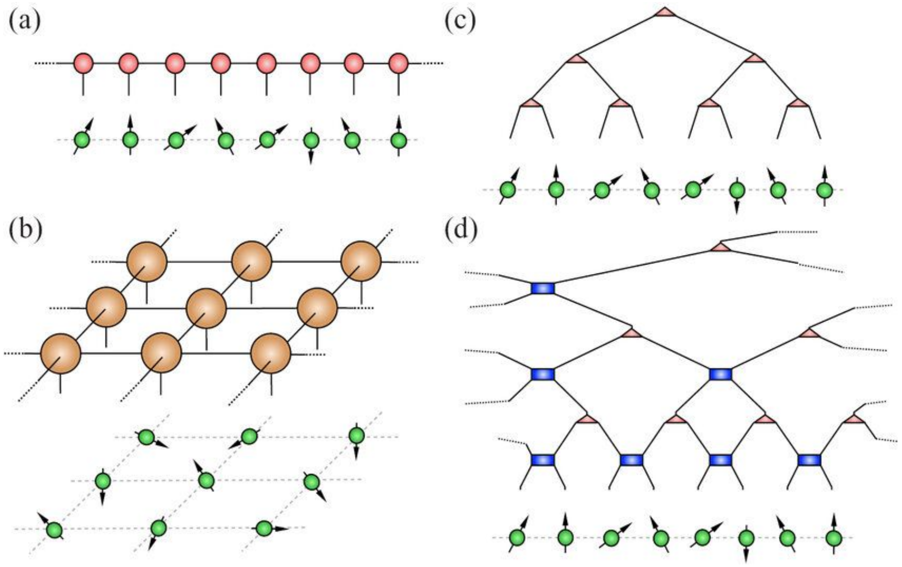
- Between ~1985-2000 most of the foundational work was done on fault-tolerant quantum algorithms
- These have provable advantages over classical algorithms in all cases.
  - Quantum Phase Estimation
  - Grover's Search
- Full list - Quantum Algorithm Zoo
  - https://quantumalgorithmzoo.org/

Asymptotic Scaling

- How much time or memory an algorithm takes as a function of the input size.
- Denoted with O-notation.
  - O(1) →Constant →Accessing a single item in a list. →N = ∞
  - O(log N) →Logarithmic →Binary search of a tree-structure. →N = Really Big
  - O(N) → Linear →Operating on all items in a list. →N = Really Big
  - O(N log N) → Polylog →Sorting items in a list. →N = Pretty Big
  - $O(N^2)$ → Quadratic →Operating on all pairs of items in a list →$N < 10^6$
  - $O(2^N)$ → Exponential →Operating on all combinations of elements in a list →N < 40
  - O(N!) → Factorial →Operating on all permutations of items in a list →N < 12
- The goal is to use quantum hardware to reduce the asymptotic scaling of a problem.

Physics Simulations

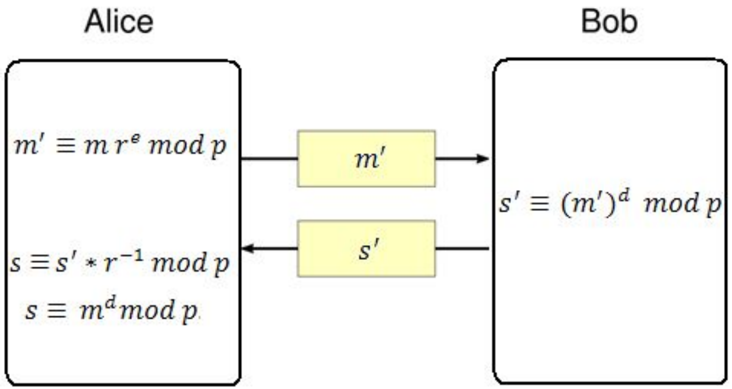
- Like we've seen, quantum Hamiltonian simulations can scale as $O(2^N)$.
- This assumes all-way entanglement.
  - Singles
  - Pairs
  - Triples
  - Triples with pairs
  - Pairs with singles
  - …
- In practice, this isn't always the case.
  - Entanglement in real systems tends to have more defined structure.
  - Quantum has asymptotic advantage where the structures still scale poorly.

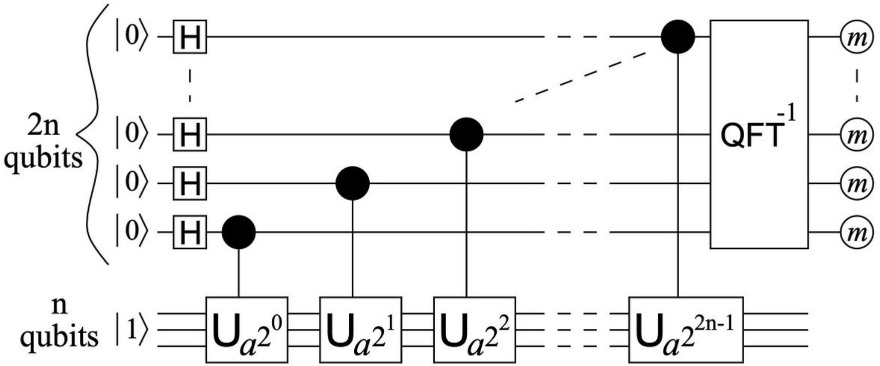


Efficiently Representable Entanglement Structures

Factoring Integers

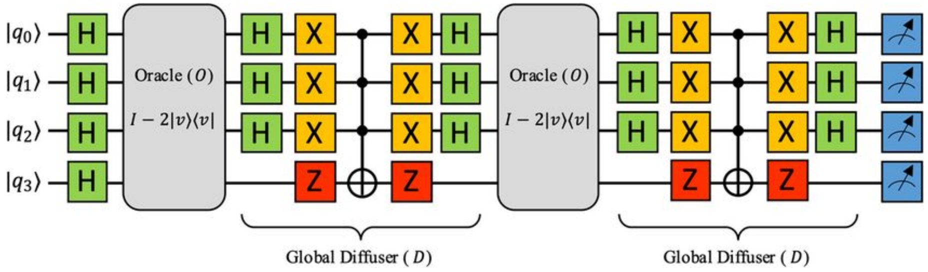
- The RSA cryptosystem is secured by the fact that it is very hard to factor an integer into its prime components. $O(2^N)$
- The factoring problem has hidden periodic structure that is quantumly easy to solve. $O(N^k)$
- Shor's algorithm exploits this structure.

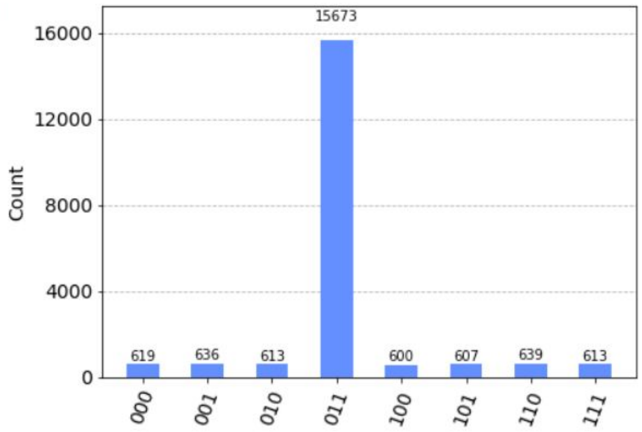




Grover Search

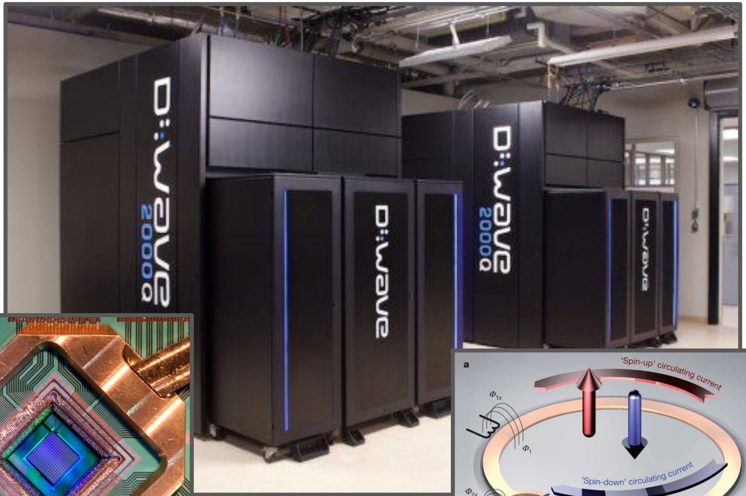
- Searching an unstructured list of items is O(N)
  - Need to check every item in the worst-case.
- Grover search reduces this to O(√N)
  - Use quantum entanglement to amplify the probability amplitude of the target item.
  - Iterate multiple times, marking and amplifying until the probability is strongly peaked around the correct item.
- *Any* classical algorithm can be converted to a Grover search problem.
  - This is why we say quantum computing guarantees a quadratic speedup.
  - Ignores “constant factors”
  - It can take a lot of work to transform the problem, washing out any asymptotic advantages except for very big values of N.

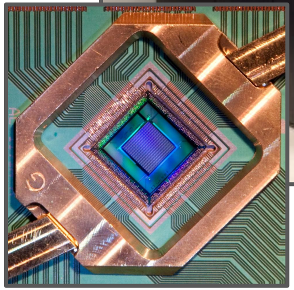
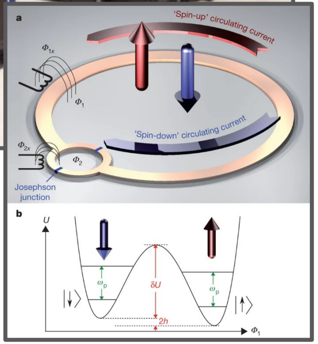


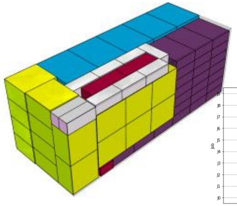


Optimization

- Consider problems where we need to find the best combination of items, paths or arrangements in a system.
  - This is the entire field of logistics and operations research.
  - Is a substantial part of industrial engineering.
- In the worst case, these problems are O(N!)
- Quantum does not guarantee an exponential speedup.
  - Can provide problem-specific advantages
- Some productive use is happening today.
  - D-Wave superconducting quantum annealers
  - QAOA algorithm

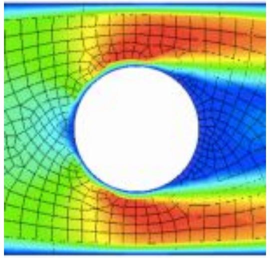


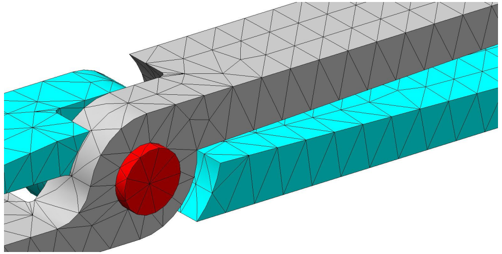



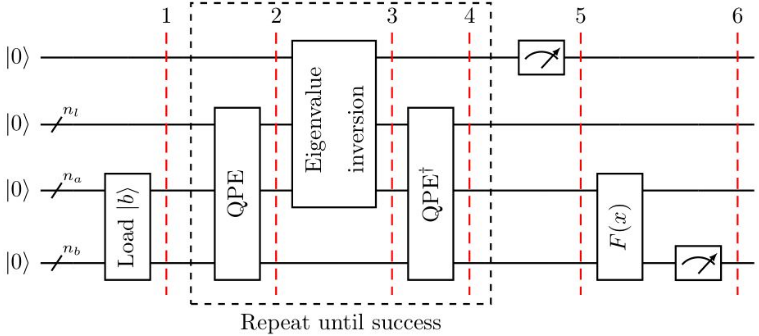




Matrix Solving

- All of engineering calculations.
  - FEA
  - CFD
  - Any system of partial differential equations
- Harrow-Hassidim-Lloyd (HHL) algorithm
  - Solves Ax=B in O(log N)!
  - Best for sparse matrices!
- The fine print:
  - It’s O(N) just to write down the problem.
  - You don’t get the solution field, only an expectation value.
  - Needs careful thought on how it’s integrated.

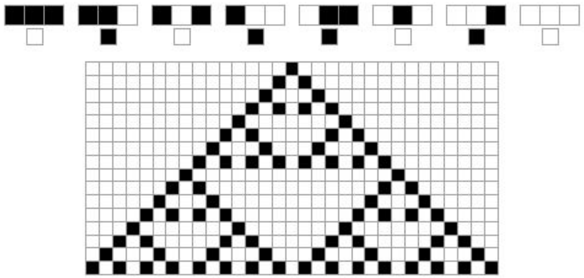


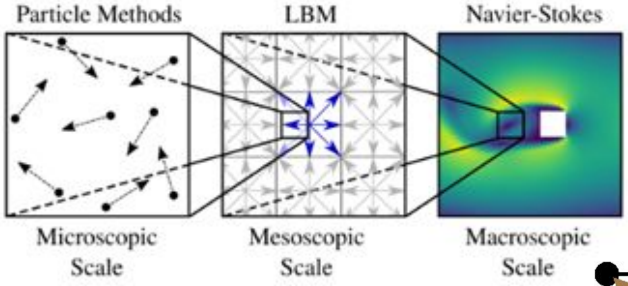


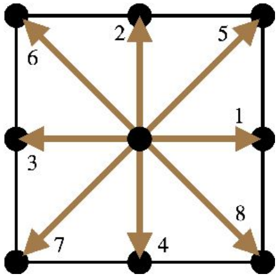


Cellular Automata & Lattice Boltzmann Methods

- Cellular automata are grid worlds of cells.
  - They update their states according to their neighbors’ states
  - Popularized by Stephen Wolfram (Mathematica, Wolfram Alpha, ...)
- A continuous value version called the Lattice Boltzmann Method is used in fluid simulations.
  - Massively parallelizable
  - Very promising for fluid-structure interactions
- Quantum versions give varying speedups
  - Problem-dependent
  - CFD problems tend to scale with the Reynolds number
- 1D demos are possible on today’s hardware.







Machine Learning

- Quantum machine learning has been demonstrated on small systems.
  - Typically train the parameters of a variational circuit like VQE
- Advantages in compactness.
  - Fewer parameters than classical models.
  - Fewer training iterations (in theory)
- Unlikely to look the same as classical ML
  - Small input data
  - Small output data