SC22
Dallas, TX | hpc accelerates.

Introduction to Quantum Computing

Scott Pakin, *Los Alamos National Laboratory*
Eleanor G. Rieffel, *NASA Ames Research Center*

13 November 2022

LA-UR-19-31426

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

All of Quantum Computing on One Slide

- **The good**
  - 2^n-way parallelism from n qubits
  - Possibility of exponential speedup for some problems
  - Some classically intractable problems can be made tractable
  - Some tractable problems can be solved asymptotically faster
  - Some problems can be solved exactly in the time it would take classically to solve them only probabilistically
- **The bad**
  - Quantum computation is extremely I/O bottlenecked: only n bits of input and n bits of output relative to 2^n-way parallelism
    - Can you think of a problem that reads a single 32-bit number, performs sequences of 4,294,967,296 concurrent operations on that number, and writes a single 32-bit number?
  - Limited applicability—note the use of "*some* problems" above
  - Programming is extremely difficult: requires expertise in linear algebra, computer science, and quantum physics as well as knowledge of prior algorithms and innate creativity
- **The ugly**
  - Contemporary quantum computers provide too few qubits even to represent most interesting problems
  - Current qubit quality is extremely low: unlikely to produce correct answers for more than handful of qubits running for more than a handful of time steps

Quantum-Computing Fundamentals

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

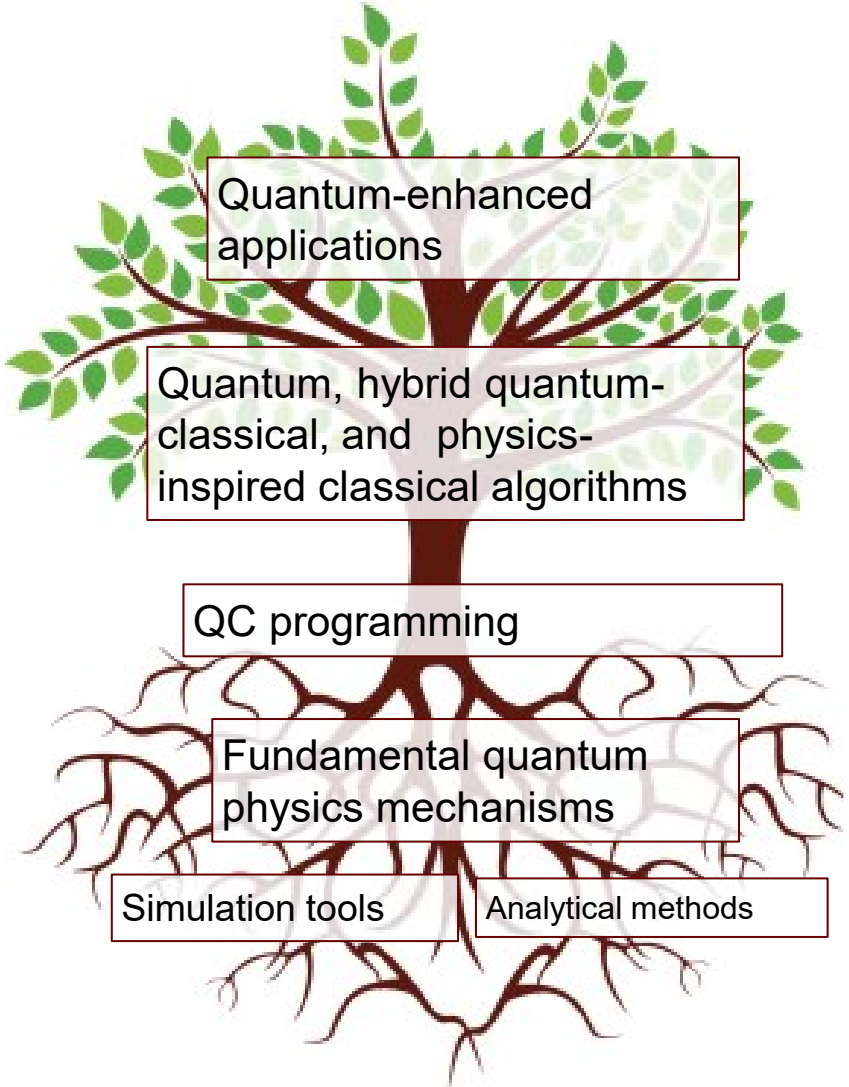
NASA's Stake in Quantum Computing

NASA constantly confronting massively challenging computational problems

- Computational capacity limits mission scope and aims

NASA's Pleiades
One of the top 25 fastest supercomputers in the world

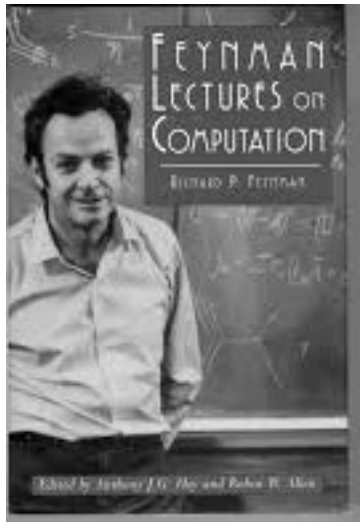
NASA QuAIL mandate:
Determine the potential for quantum computation to enable ***more ambitious NASA missions*** in the future

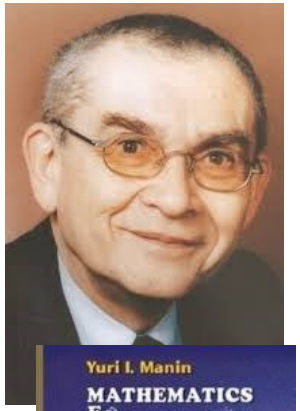


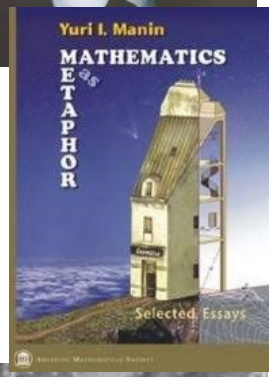
NASA Ames QuAIL team

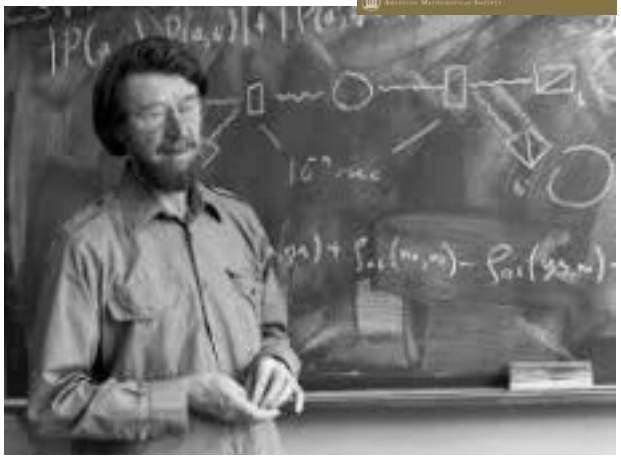
Birth of Quantum Computing

- **Feynman and Manin recognized in the early 1980s that certain quantum phenomena could not be simulated efficiently by a computer**
  - Phenomena related to quantum entanglement; Bell's inequality
- **Perhaps these quantum phenomena could be used to speed up more general computation?**



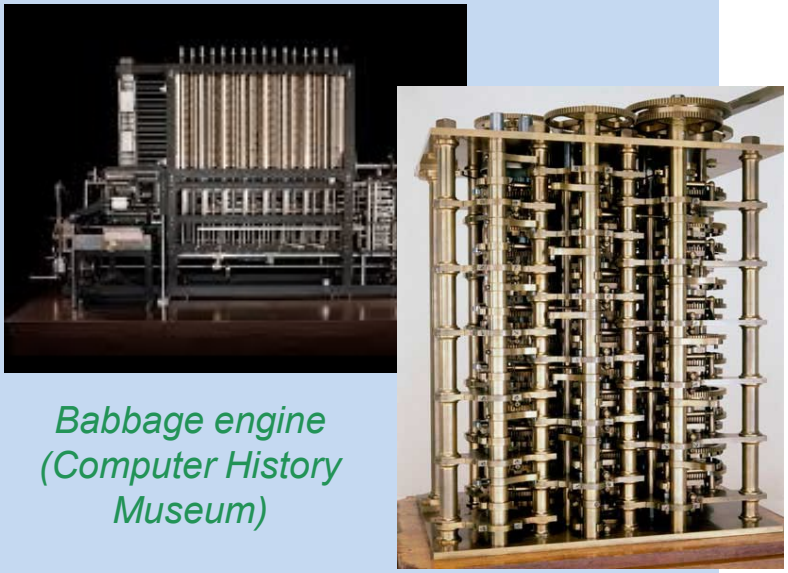






Computers as Classical Mechanical Machines

- **Babbage's analytical engine was a classical mechanical machine**
- **Turing machines**
  - The abstraction that underlies complexity theory and universal computing machines
  - Firmly rooted in classical mechanics
  - Described in classical mechanical terms
- **Abstraction allowed us ignore how classical computers are implemented physically**
  - When we program we don't think about the fundamental physics
- ***How do different models of physics affect how quickly we can compute?***



Babbage engine (Computer History Museum)

Computers as Quantum Mechanical Machines?

Fundamental questions

- **How do different models of physics affect how quickly we can compute?**
  - Suggests new computation-based physics principles
- **How would basing computation on a quantum mechanical model rather than a classical mechanical model change our notions of computing?**
  - Quantum physics is the physics of our universe
- **How quickly does nature allow us to compute?**

What a Quantum Computer is Not

- **Just because a computer uses quantum effects, does not mean it is a quantum computer**
  - All the computers in this building make use of quantum effects
  - The fundamental unit of computation, the bit, and the algorithms we design for computers did not change when quantum effects were used
- **A quantum computer has a fundamentally different way of encoding and processing information**
  - Quantum computers are quantum information processing devices
  - They process qubits instead of bits
  - They use quantum operations instead of logic gates
- **Also, just because a piece of hardware has a certain number of qubits, it isn't necessarily a quantum computer**
  - A set of light switches, even a very large set, is not a classical computer

Certainty and Randomness in Quantum Computation

- **Any computation a classical computer can do, a quantum computer can do with roughly the same efficiency**
  - With the same probability of the outcome
  - If the classical computation is non-probabilistic, so is the quantum one
- **Like classical algorithms, some quantum algorithms are inherently probabilistic and others are not**
  - First quantum algorithms were not probabilistic
    - E.g. Deutsch-Jozsa algorithm solves problem with certainty that classical algorithms, of equivalent efficiency, could solve only with high probability
  - Shor's algorithms are probabilistic
  - Grover's is not intrinsically probabilistic
    - initial search algorithm was probabilistic, but
    - slight variants, which preserve the speed up, are non-probabilistic

Current Status of Quantum Algorithms

Quantum computing can do everything a classical computer can do

and

Provable quantum advantage known for a few dozen quantum algorithms

Unknown quantum advantage for everything else

Status of classical algorithms

- Provable bounds hard to obtain
  - Analysis is just too difficult
- Best classical algorithm not known for most problems
- Empirical evaluation required
- Ongoing development of classical heuristic approaches
  - Analyzed empirically: ran and see what happens
  - E.g. SAT, planning, machine learning, etc. competitions

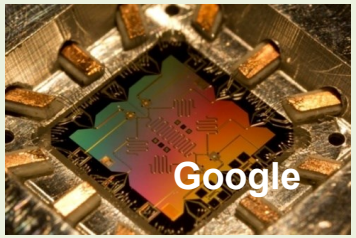
- **NISQ era supports unprecedented means for empirical analysis of quantum algorithms**
  - **Quantum heuristics come into their own**

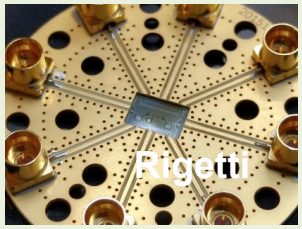
A handful of proven limitations on quantum computing

Conjecture: Quantum Heuristics will significantly broaden applications of quantum computing

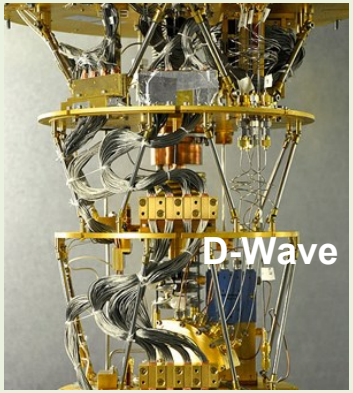
Quantum Hardware

General Purpose:
Universal quantum processors





Special Purpose:
E.g. Quantum annealers



Noisy Intermediate-Scale Quantum (NISQ) devices

Superconducting quantum processors

Trapped ion quantum processors

Photonic quantum processors

Other approaches
- **Electron spins in silicon**
- **Neutral atom, cold atom**
- **Topological, anyon based quantum computing**

Number of qubits alone is not a good measure
- **Analogy: billions of switches do not a classical computer make**

Other key factors
- **precision, speed, and generality of the control**
  - **particularly operations involving multiple qubits**
- **how long quantum coherence can be maintained**
- **stability over time**
- **speed with which processors can be calibrated**

Quantum Computing has Entered the NISQ Era

Quantum supremacy has been achieved!

- Perform computations not possible on even the largest supercomputers in a reasonable amount of time

Cover article, Nature, 24 Oct 2019


Article

Quantum supremacy using a programmable superconducting processor

Google, NASA, ORNL collaboration

https://www.nature.com/articles/s41586-019-1666-5

https://www.nasa.gov/feature/ames/quantum-supremacy

… but not <u>useful</u> quantum supremacy.

- Currently too small to be useful for solving practical problems
- Perhaps an early application to certified random number generation, but other applications require larger, more capable devices

Uses of these still limited, quantum devices?

(1) Unprecedented opportunity to explore and **evaluate algorithms**, both quantum and hybrid quantum-classical heuristic algorithms

(2) Investigate quantum mechanisms that may be harnessed for computational purposes

Insights gained feed into next generation

- **quantum algorithms**
- **quantum hardware**

Early target: Optimization, Machine Learning, Chem & Materials Simulation

Three Group Exercises

- **Before going on to a more technical part introducing the fundamentals of quantum computation**

Exercise 1

- **Which of the following best describes the current status of quantum algorithms?**
  a) Quantum algorithms can beat classical algorithms on every problem, we just need to build quantum computers on which to run them!
  b) While there are only a few dozen quantum algorithms known, quantum algorithms continue to be discovered, with many more algorithms likely to be identified as larger processors are built, enabling the evaluation of quantum heuristics.
  c) Quantum algorithms have been studied since the early 1990s, and pretty much everything is known by now.
  d) Quantum mechanics is the physics of the universe. Every algorithm is a quantum algorithm!

Exercise 2

- **Which statement best describes “quantum supremacy”?**
  a) “Quantum supremacy” was already achieved in the 1990s by Shor’s algorithm, since it is a polynomial time algorithm whereas the best classical algorithms are superpolynomial time algorithms.
  b) It is well-known that quantum computers can beat classical computers, even supercomputers, at everything. “Quantum supremacy” is just a quick way of saying that.
  c) A quantum processor demonstrating “Quantum supremacy” means it has been able to perform in a practical amount of time a computation that could not be performed on even the world’s largest supercomputers in a practical amount of time. It would be achieved even if it was demonstrated for only one computation and that computation was useless.
  d) “Quantum supremacy” will be achieved only when quantum computers can run Shor’s algorithm on cryptographically relevant numbers.

Exercise 3

- **Which statement best describes the relation between uncertainty and quantum algorithms?**
  a) Like classical algorithms, quantum algorithms fall in two categories, algorithms that provide an answer with certainty and probabilistic algorithms
  b) Quantum mechanics is by nature uncertain—think the quantum uncertainty principle—so unlike classical algorithms, quantum algorithms are inherently probabilistic
  c) Classical algorithms can be translated to a form that can be run on quantum computers, so translations of classical algorithms that answer with certainty, still answer with certainty, but if an algorithm makes use of truly quantum effects, it cannot provide an answer with certainty
  d) All algorithms, both quantum and classical, cannot provide a result with certainty—life is inherently uncertain.

Agenda

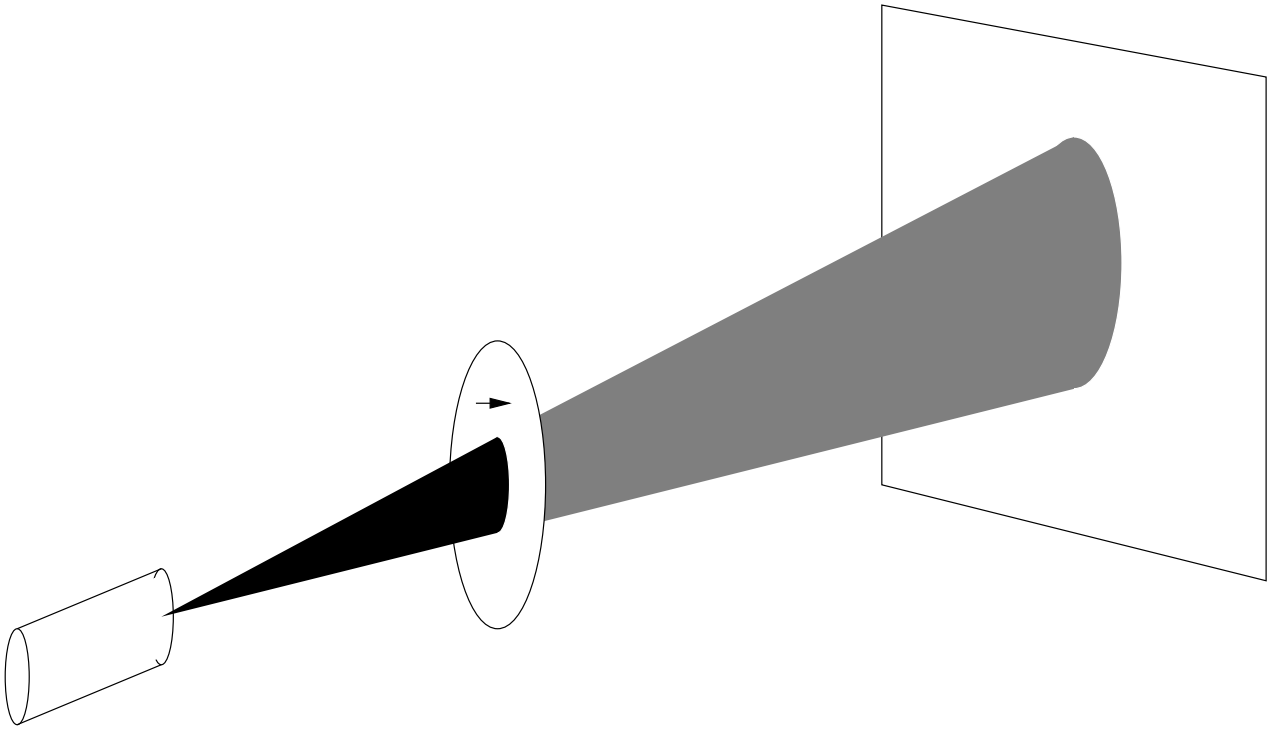
- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

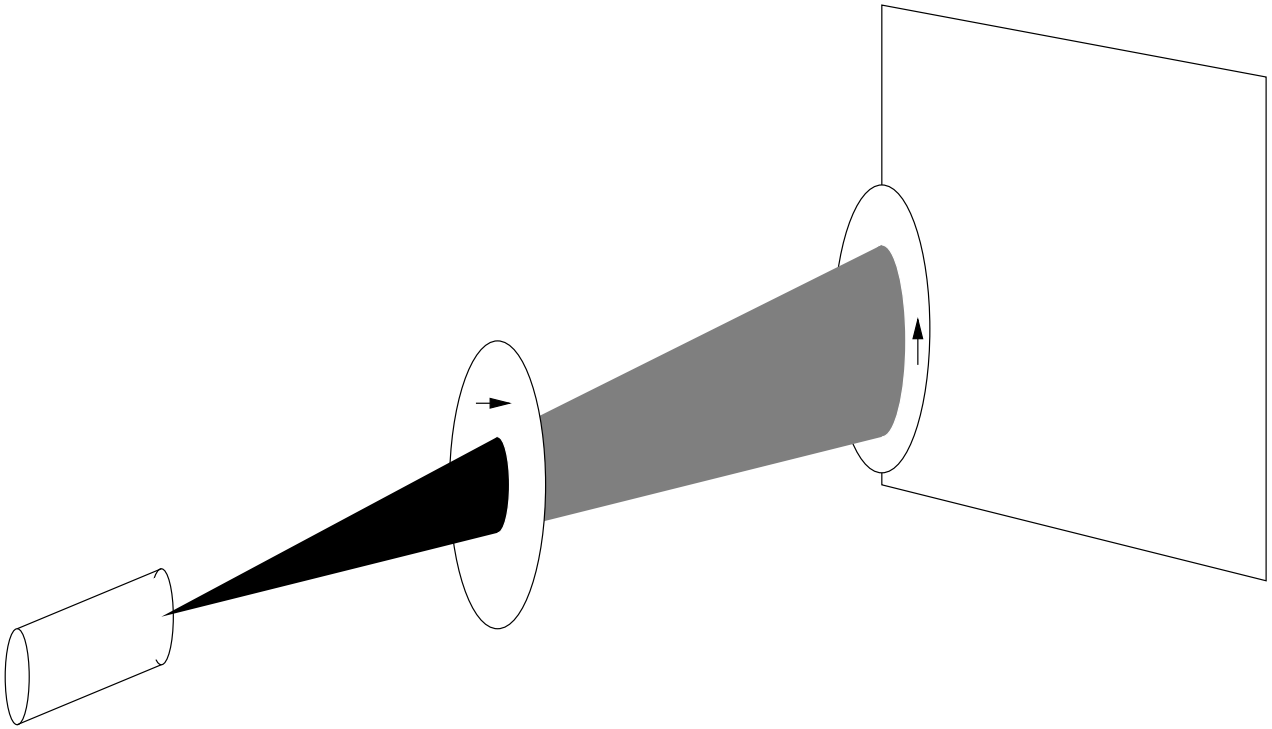
- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

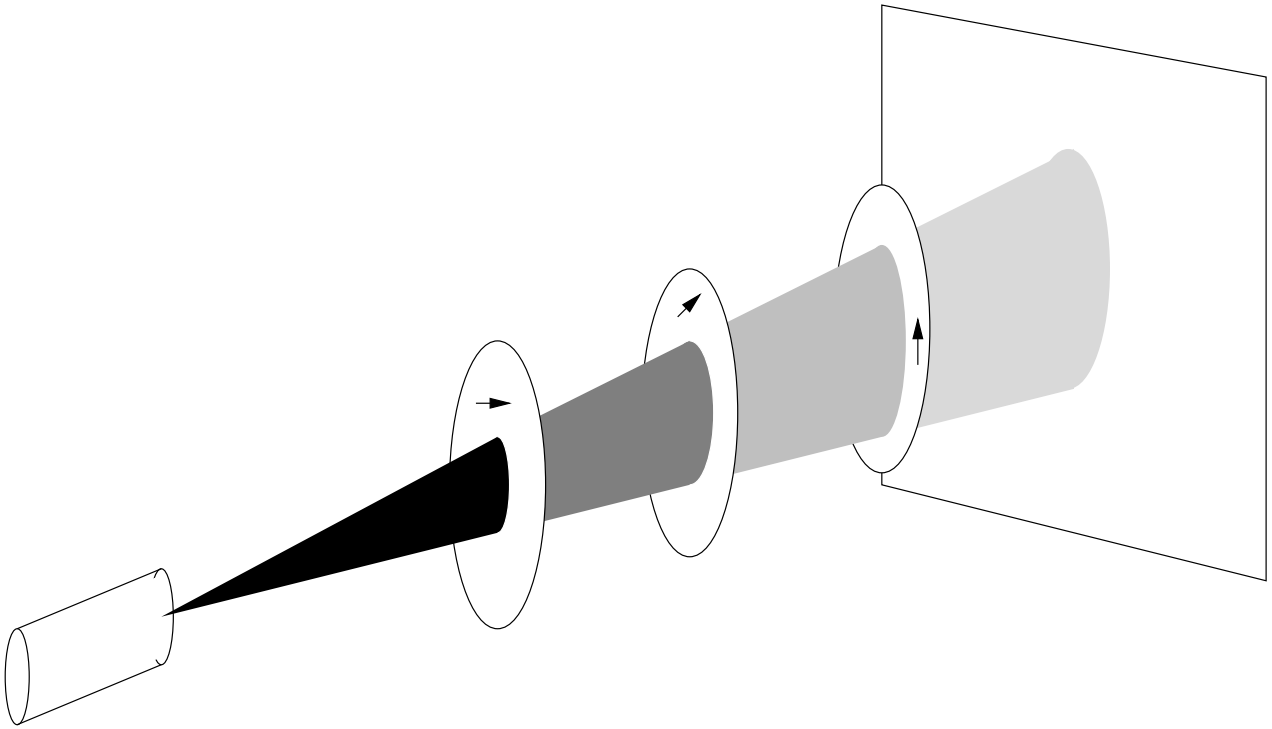
A Simple Experiment: Photon Polarization



A Simple Experiment: Photon Polarization



A Simple Experiment: Photon Polarization



Mathematically Representing Photon Polarization

Polarization state of a photon

- can be represented as a 2-dimensional vector of unit length

Taking horizontal $|\rightarrow\rangle$ and vertical $|\uparrow\rangle$ polarizations as a basis, an arbitrary polarization can be expressed as a superposition

$$|\psi\rangle = a|\uparrow\rangle + b|\rightarrow\rangle$$

with $|a|^2 + |b|^2 = 1$

(Allowing a and b to be complex numbers enables this formalism to describe circular polarization as well)

$|v\rangle$ is Dirac's notation for vectors. Means the same thing as $\vec{v}$ or $\mathbf{v}$, with v being the label for the vector

Measurement of Polarization

Polarization filters are quantum measuring devices

Quantum measurements always occur w.r.t. an orthogonal subspace decomposition associated with the measuring device

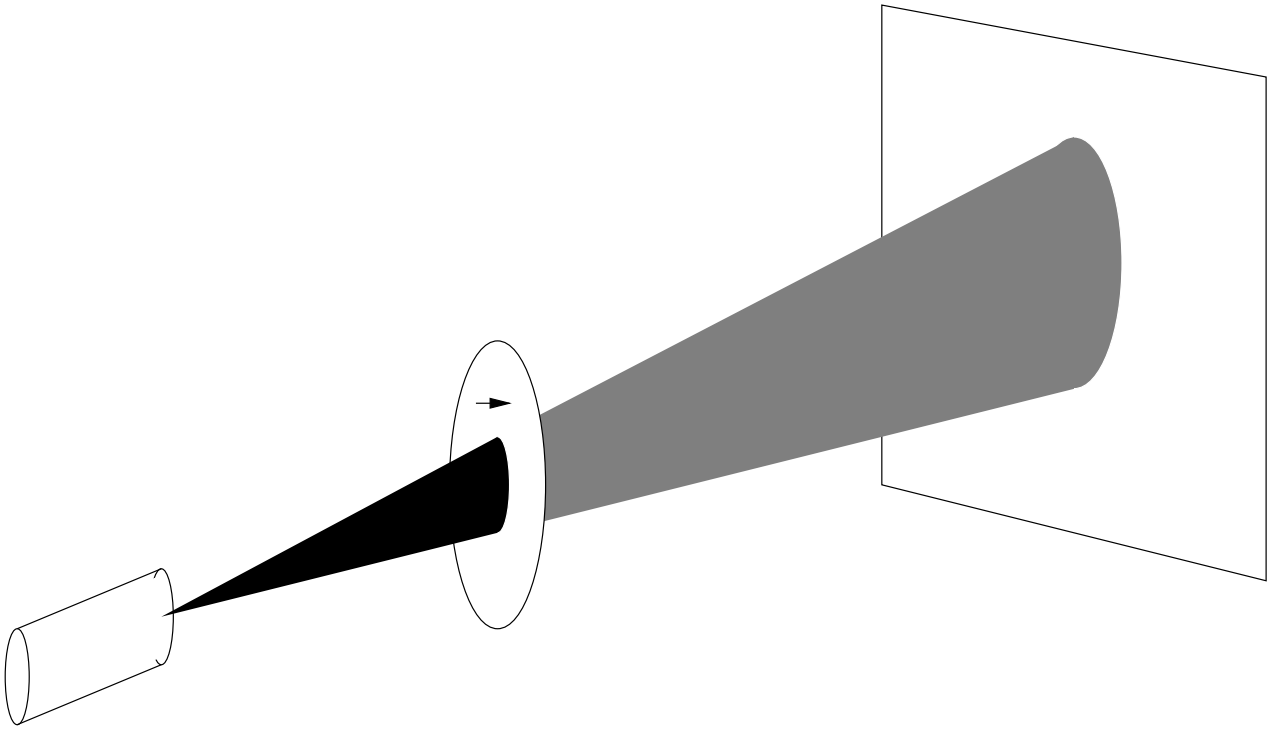
For a horizontal polarization filter, the basis in which it measures is $|\rightarrow\rangle$, together with its perpendicular $|\uparrow\rangle$

A photon with polarization $a|\uparrow\rangle + b|\rightarrow\rangle$ is measured by a horizontal filter as $|\uparrow\rangle$ (absorbed) with probability $|a|^2$, and
$|\rightarrow\rangle$ (passed) with probability $|b|^2$

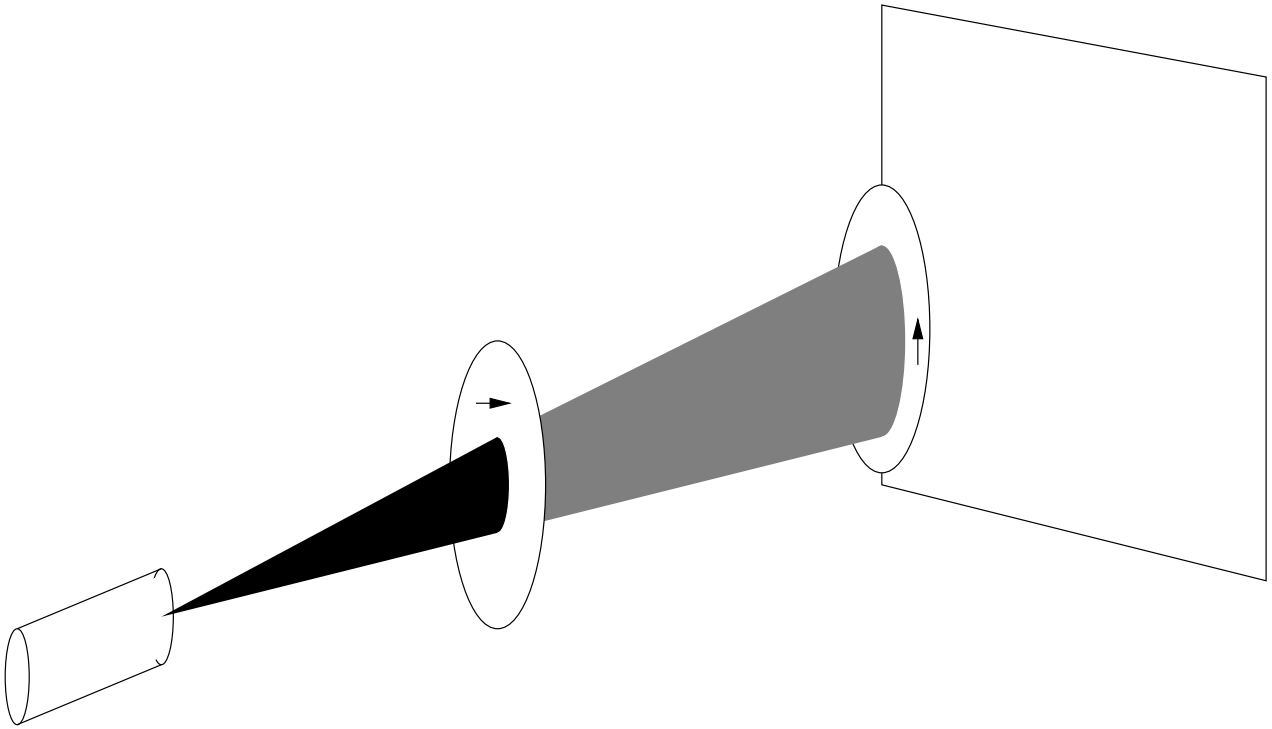
Any photon that has passed through the filter now has polarization $|\rightarrow\rangle$.

Polarization filters at other angles work in a similar way

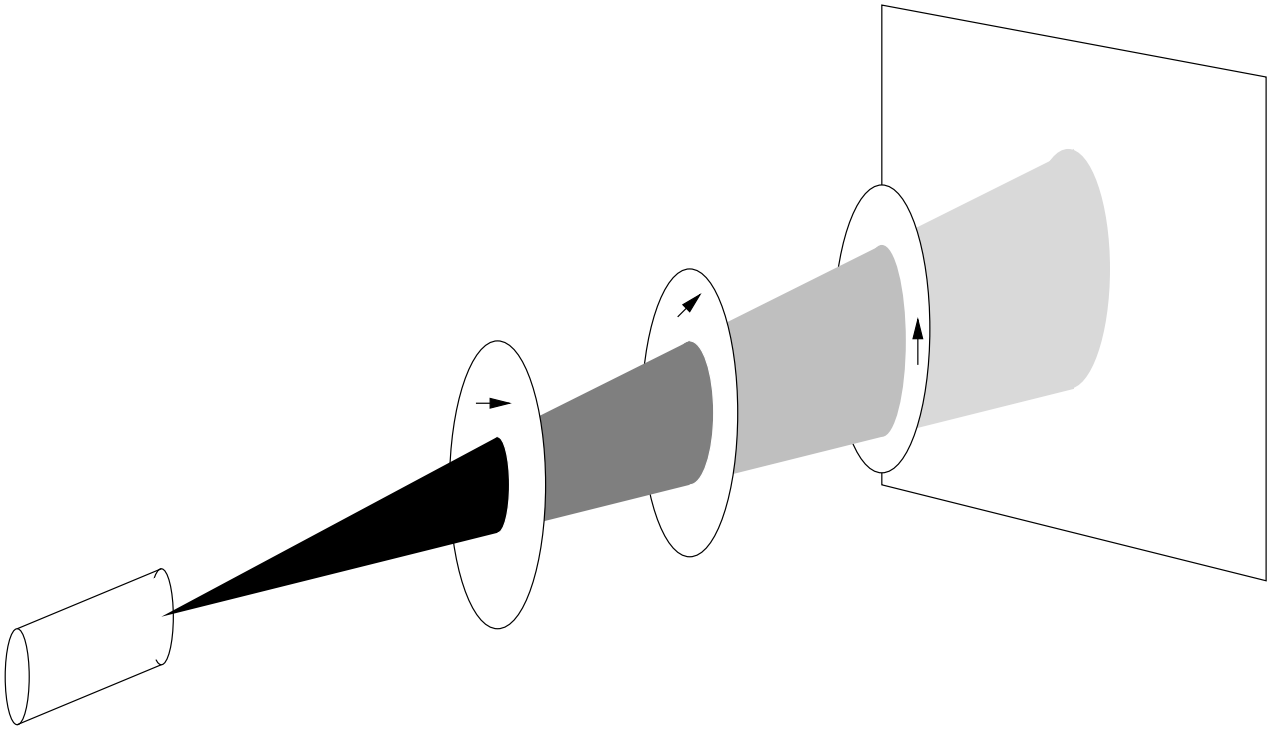
The Photon Polarization Experiment Revisited



The Photon Polarization Experiment Revisited



The Photon Polarization Experiment Revisited



Qubits (Quantum Bits)

Think polarization states of a photon!

Any 2-dimensional quantum system can be viewed as the fundamental unit of quantum computation, a *quantum bit* or *qubit*.
Qubit state space is a 2-dimensional complex vector space

A computational basis is chosen, denoted $|0\rangle$ and $|1\rangle$, and used to encode classical bit values 0 and 1

Possible qubit values $a|0\rangle + b|1\rangle$, for complex a, b with $|a|^2 + |b|^2 = 1$.

Unlike classical bits, qubits can be in superposition states such as $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or $\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$

Measurement of Single Qubits

Measuring qubit $a|0\rangle + b|1\rangle$ in the computational basis $\{|0\rangle, |1\rangle\}$

- returns 0 with probability $|a|^2$
- returns 1 with probability $|b|^2$
- projects to state to the basis state corresponding to the measurement result

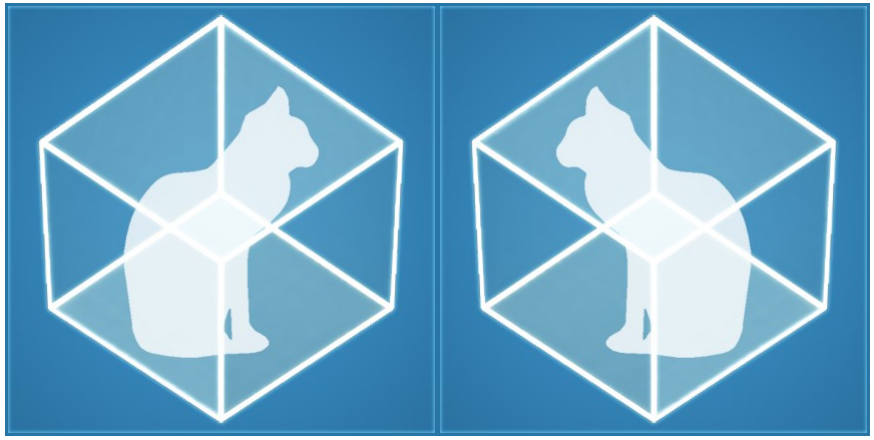
A qubit can be measured with respect to any orthogonal basis for its 2-dimensional state space

Only one classical bit of information can be extracted from one qubit

No cloning theorem: An unknown quantum state cannot be reliably copied

Multiple Qubits

- Qubits combine like quantum particles not classical objects
  - Quantum states combine via tensor products not direct products
  - The quantum state space, the space of possible states of n quantum particles, is exponentially larger than that of n classical objects
  - 2^n instead of 2n
- Entangled states make up the bulk of this space
- No classical analog: The state of entangled multiple particle systems cannot be described in terms of the states of the individual particles



High-level View of How State Spaces Combine

Let X be a vector space with basis $\{|\alpha_1\rangle, \ldots, |\alpha_n\rangle\}$ and Y be a vector space with basis $\{|\beta_1\rangle, \ldots, |\beta_m\rangle\}$

Classical state spaces combine via the **Cartesian product**

$X \times Y$ has basis
$\{|\alpha_1\rangle, \ldots, |\alpha_n\rangle, |\beta_1\rangle, \ldots, |\beta_m\rangle\}$

$$\begin{aligned} \dim(X \times Y) &= \dim(X) + \dim(Y) \\ &= n + m \end{aligned}$$

Quantum state spaces combine via the **tensor product**

$X \otimes Y$ has basis
$\{|\alpha_1\rangle \otimes |\beta_1\rangle, |\alpha_1\rangle \otimes |\beta_2\rangle, \ldots, |\alpha_n\rangle \otimes |\beta_m\rangle\}$

$$\begin{aligned} \dim(X \otimes Y) &= \dim(X) * \dim(Y) \\ &= n * m \end{aligned}$$

Scott will go over the mathematics and notation here in more detail in the next segment of the tutorial.

Exponential State Space

The quantum state of an n qubit system is a vector in a 2^n-dimensional space

If B is the state space of a single qubit spanned by $\{|0\rangle, |1\rangle\}$, then a 2-qubit system $B \otimes B$ has basis

$$\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\},$$

often written

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\},$$

The standard computational basis for the 2^n-dimensional complex vector space $B \otimes B \ldots B \otimes B$ of an n qubit system is

$$\{|00\ldots00\rangle, |00\ldots01\rangle, \ldots, |11\ldots10\rangle, |11\ldots11\rangle\}$$

We'll use the notation $|5\rangle = |101\rangle$ when n is understood.

Quantum vs. classical state spaces

A general n-qubit state can be written as

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

where $\sum_i |\alpha_i|^2 = 1$

Since $2n \ll 2^n$, most n-qubit states cannot be described by the states of the individual qubits

- Most states **cannot** be written as the tensor product of individual qubit states
- (All states can be written as a linear combination of such states.)

States that cannot be written as the tensor product of individual qubit states are called *entangled* states

- These states have no classical counterpart

Measurement of Single Qubits

Any measuring device has an associated splitting of the 2^n–dim state space $\mathcal{H}$ into orthogonal subspaces $S_1, \ldots, S_k$ with $\mathcal{H} = S_1 \times S_2 \times \cdots \times S_k$

- The only possible outcomes of a measurement are states in one of the subspaces of the orthogonal decomposition associated with the device

Measurement is probabilistic

- Depends on the amplitude of the state in each subspace
- When the device measures a quantum state $|\psi\rangle$, one of the S_j's is chosen with probability the square of the amplitude of the component of $|\psi\rangle$ in S_j

Measurement changes the state

- To one compatible with the measurement result (in the right subspace).
- The state after measurement is the unit vector aligned with the projection of the original state onto S_j

Entangled States

Entangled states cannot be written as tensor product of independent qubits

Example: An EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

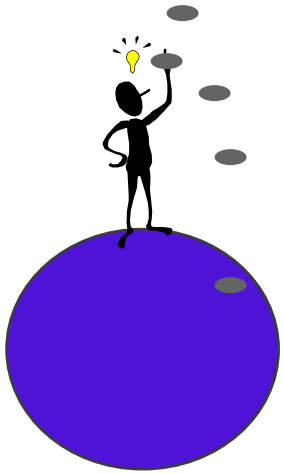
$$\begin{aligned} & (a_0|0\rangle + b_0|1\rangle) \otimes (a_1|0\rangle + b_1|1\rangle) \\ = \;& a_0 a_1|00\rangle + a_0 b_1|01\rangle + b_0 a_1|10\rangle + b_0 b_1|11\rangle \\ \neq \;& a_0 a_1|00\rangle + 0|01\rangle + 0|10\rangle + b_0 b_1|11\rangle \\ = \;& \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \end{aligned}$$

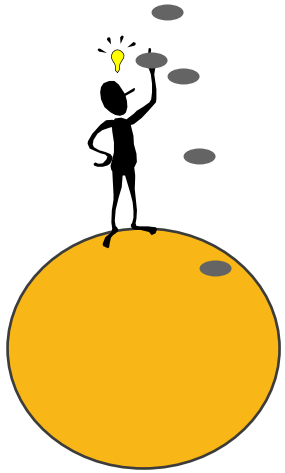
- Measurement of the first qubit yields either $|0\rangle$ or $|1\rangle$
- Measurement changes state to either $|00\rangle$ or $|11\rangle$
- Measurement of second qubit gives same result as first

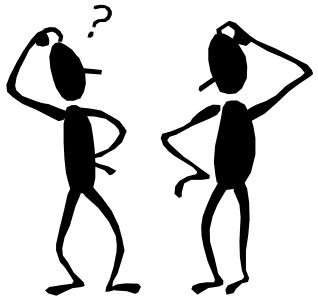
Similar results when measuring in other bases

Entanglement, correlations, and communication

- Two people each see completely random results from their coin tosses
- Completely correlated results!
  - But no way to know this unless they communicate
- There is no way to use this to communicate
  - Different relativistic frames disagree about who flipped the coin first







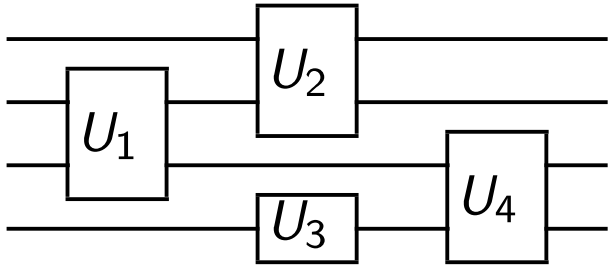
Critically important also: the behavior when they measure in different basis.

Quantum Computer (Circuit Model)

A quantum computation consists of

- initialization of n-qubit register ($|\psi\rangle$)
- quantum state transformation of register
  - sequence of primitive (1- or 2-qubit) operations (gates) U_i that collectively perform the transformation of the register
- measurement of some or all of the qubits of the register
- classical control throughout to
  - program which quantum steps to carry out
  - interpret results of quantum measurement

$$(I \otimes I \otimes U_4)(U_2 \otimes I \otimes U_3)(I \otimes U_1 \otimes I)|\psi\rangle$$



Exercise

Which of the following states

a) $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

b) $|00101\rangle$

c) $|{+}{+}\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle$

d) $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

e) $|w_4\rangle = \frac{1}{2}(|0001\rangle + |0010\rangle + |0100\rangle + |1000\rangle)$

i) are superpositions in the standard basis?
ii) are superpositions in the Hadamard basis $\{|+\rangle, |-\rangle\}$**,** where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

iii) are entangled?

- **Bonus exercise: Prove the no cloning theorem**
  (Hint: follows from linearity of quantum operations)

Notation

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

Tensor Products

- **The tensor product, $\otimes$, multiplies two vectors to produce a longer vector or two matrices to produce a larger matrix**
  - Unlike dot products or matrix multiplication, the two arguments do not have to have compatible dimensions
- **Operational semantics (loosely specified)**
  - Multiply each scalar on the left-hand-side vector/matrix by the entire right-hand-side vector/matrix
- **Vector example**

$$- \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}, \quad \text{e.g., } \begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \cdot 3 \\ 1 \cdot 4 \\ 2 \cdot 3 \\ 2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix}$$

- **Matrix example**

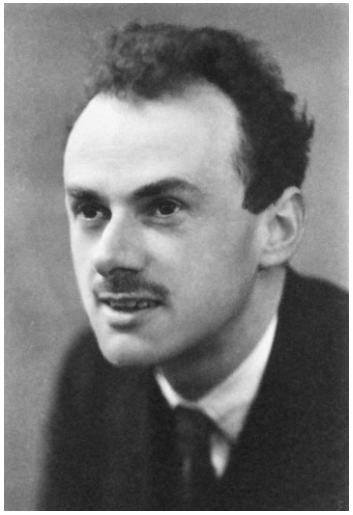
$$- \begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}, \quad \text{e.g., } \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \otimes \begin{pmatrix} 3 & 1 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 6 & 2 \\ 1 & 4 & 2 & 8 \\ 6 & 2 & 3 & 1 \\ 2 & 8 & 1 & 4 \end{pmatrix}$$

Basics of Dirac (a.k.a. Bra-Ket) Notation

- **Two components: bras and kets**

$\langle\psi|$ *"Bra"* — Row vector (adjoint)

$|\psi\rangle$ *"Ket"* — Column vector



Paul Dirac
1902–1984

- **The label (e.g., "ψ") is merely a name and has no inherent meaning**
- **However, some conventions exist:**

$$|0\rangle \equiv \begin{pmatrix}1\\0\end{pmatrix} \quad |1\rangle \equiv \begin{pmatrix}0\\1\end{pmatrix} \quad |+\rangle \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \quad |-\rangle \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

- **Bra times ket: $\langle\psi|\phi\rangle$**
  - Inner product
  - Returns a scalar
- **Ket times bra: $|\phi\rangle\langle\psi|$**
  - Outer product
  - Returns a matrix

Hence, e.g.,

$$\langle -| \equiv \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\end{pmatrix}$$

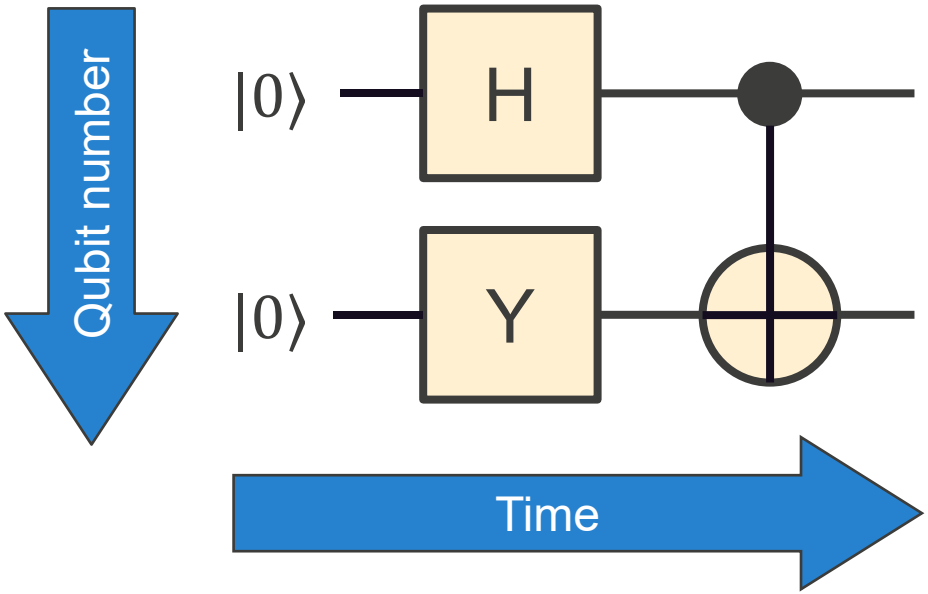
More on Dirac Notation

- **Ket-kets and bra-bras**
  - $|a\rangle|b\rangle$ includes an implicit tensor product: $|a\rangle \otimes |b\rangle$
  - We routinely simplify this even further to just $|ab\rangle$
  - Example: Given that $|0\rangle \equiv \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle \equiv \begin{pmatrix}0\\1\end{pmatrix}$, $|01\rangle = |0\rangle|1\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$
- **Simple cases**
  - Given two orthogonal kets $|\uparrow\rangle$ and $|\downarrow\rangle$ that are each normalized (this is typical), $\langle\uparrow|\uparrow\rangle = \langle\downarrow|\downarrow\rangle = 1$ and $\langle\uparrow|\downarrow\rangle = \langle\downarrow|\uparrow\rangle = 0$
- **Convenient way to reason about linear transformations**
  - $|\text{out}\rangle\langle\text{in}|$ is an operator that maps $|\text{in}\rangle$ to $|\text{out}\rangle$ (i.e., by left multiplication) and anything orthogonal to $|\text{in}\rangle$ to a zero vector: $|\text{out}\rangle\langle\text{in}|\text{in}\rangle = |\text{out}\rangle$; $|\text{out}\rangle\langle\text{in}|\text{out}\rangle = \mathbf{0}$
- **Distributive properties**
  - Example: assuming $|x\rangle \perp |y\rangle$ and $\| |x\rangle \| = \| |y\rangle \| = 1$, $(|x\rangle\langle y| - i|y\rangle\langle x|)\ |x\rangle = |x\rangle\langle y|x\rangle - i|y\rangle\langle x|x\rangle = |x\rangle \cdot 0 - i|y\rangle \cdot 1 = -i|y\rangle$
  - Also, $|a\rangle\langle b| \otimes |c\rangle\langle d| = |ac\rangle\langle bd|$

Examples of Working with Dirac Notation

- **Let $|w\rangle \equiv \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ in the following examples**
- **Example 1 (bra-ket): Evaluate $\langle w|w\rangle$**
  - $\langle w|w\rangle = \left(\frac{1}{2}\langle 0| + \frac{\sqrt{3}}{2}\langle 1|\right)\left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right) = \frac{1}{4}\langle 0|0\rangle + \frac{\sqrt{3}}{4}\langle 0|1\rangle + \frac{\sqrt{3}}{4}\langle 1|0\rangle + \frac{3}{4}\langle 1|1\rangle = \frac{1}{4}\cdot 1 + \frac{\sqrt{3}}{4}\cdot 0 + \frac{\sqrt{3}}{4}\cdot 0 + \frac{3}{4}\cdot 1 = 1$
- **Example 2 (ket-bra): Expand $|w\rangle\langle w|$**
  - $|w\rangle\langle w| = \left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right)\left(\frac{1}{2}\langle 0| + \frac{\sqrt{3}}{2}\langle 1|\right) = \frac{1}{4}|0\rangle\langle 0| + \frac{\sqrt{3}}{4}|0\rangle\langle 1| + \frac{\sqrt{3}}{4}|1\rangle\langle 0| + \frac{3}{4}|1\rangle\langle 1|$
- **Example 3 (operator-ket): Apply $|w\rangle\langle w|$ to $|w\rangle$**
  - Hard way: $(|w\rangle\langle w|)|w\rangle = \left(\frac{1}{4}|0\rangle\langle 0| + \frac{\sqrt{3}}{4}|0\rangle\langle 1| + \frac{\sqrt{3}}{4}|1\rangle\langle 0| + \frac{3}{4}|1\rangle\langle 1|\right)\left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right) = \left(\frac{1}{8}|0\rangle + 0 + \frac{\sqrt{3}}{8}|1\rangle + 0\right) + \left(0 + \frac{3}{8}|0\rangle + 0 + \frac{3\sqrt{3}}{8}|1\rangle\right) = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$
  - Easy way: $|w\rangle(\langle w|w\rangle) = |w\rangle \cdot 1 = |w\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$
- **Example 4 (ket-ket): Expand $|ww\rangle$**
  - $|ww\rangle = |w\rangle|w\rangle = \left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right)\left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right) = \frac{1}{4}|00\rangle + \frac{\sqrt{3}}{4}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{3}{4}|11\rangle$

The Circuit Model of Quantum Computing



- **A labelled box represents single-qubit operators (2×2 matrix)**
- **Symbol–vertical line–symbol represents a two-qubit operator (4×4 matrix)**
- **A quantum circuit is really just a piecewise representation of an enormous unitary matrix ($2^n \times 2^n$ for an *n*-qubit system)**

– Above: $(CNOT)(H \otimes Y) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & -i & 0 & -i \\ i & 0 & i & 0 \\ i & 0 & -i & 0 \\ 0 & -i & 0 & i \end{pmatrix}$

Mathematical Forms Commonly Encountered in QC

- **Magnitude of a complex number, $|\cdot|$**
  - $|a+bi| \equiv \sqrt{a^2+b^2}$
- **Vector and matrix adjoint, $A^\dagger$**
  - Complex-conjugate transpose
  - $\begin{pmatrix} a+bi \\ c+di \end{pmatrix}^\dagger \equiv \begin{pmatrix} a-bi & c-di \end{pmatrix}$
  - $\begin{pmatrix} a+bi & c+di \\ e+fi & g+hi \end{pmatrix}^\dagger \equiv \begin{pmatrix} a-bi & e-fi \\ c-di & g-hi \end{pmatrix}$
- **Matrix types**
  - Hermitian: $A = A^\dagger$
  - Unitary: $A^\dagger A = AA^\dagger = I$
- **Matrix exponentials**
  - For square matrix A, $e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$
  - In the above, $A^0 \equiv I$ for the I with the same dimensions as A

The Circuit Model

Agenda

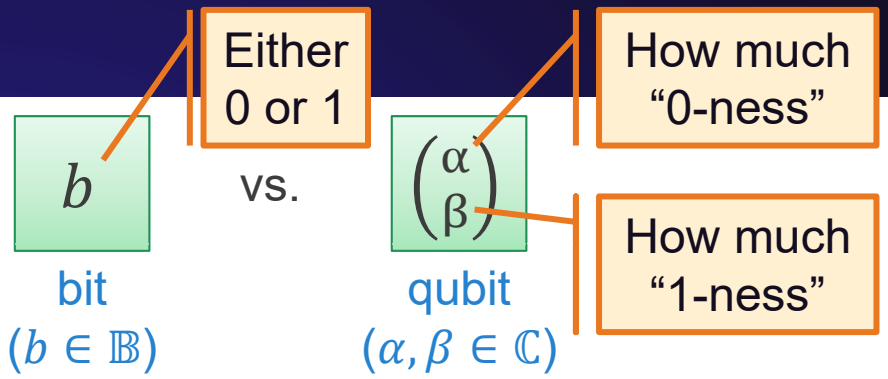
- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

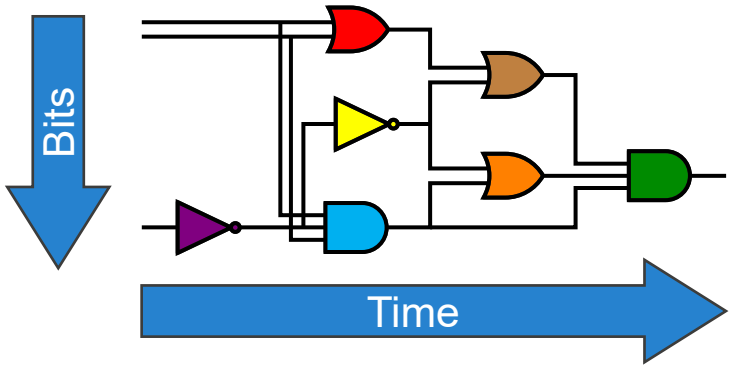
Reminders

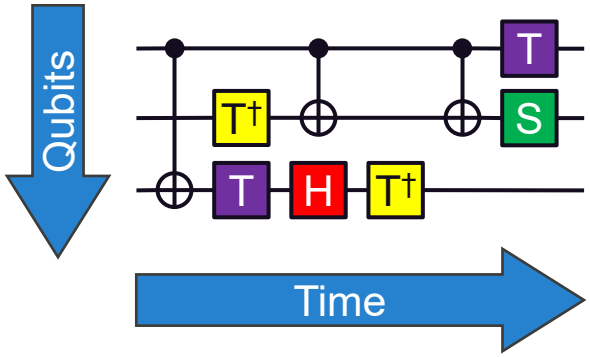


- **Unit of information**
  - Classical: Single bit, b
  - Quantum: Complex 2-vector, $|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$
- **Measurement**
  - Measuring a qubit forces it to either 0 or 1
- **Superposition**
  - If qubit $|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha\begin{pmatrix}1\\ 0\end{pmatrix} + \beta\begin{pmatrix}0\\ 1\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$, then it will be measured as 0 with probability $|\alpha|^2$ and as 1 with probability $|\beta|^2$
- **Multiple-qubit representation**
  - A two-qubit state is a complex 4-vector $|pq\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = \begin{pmatrix}\alpha\\ \beta\\ \gamma\\ \delta\end{pmatrix}$ (00, 01, 10, 11)
  - An *n*-qubit state is a complex 2^n-vector
- **Entanglement**
  - The qubits in a two-qubit state are *entangled* if they can't be factored into $|p\rangle \otimes |q\rangle$
  - Example: $\frac{1}{2}(1 \quad -1 \quad 1 \quad -1)^\mathsf{T}$ can be factored into $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\ 1\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ -1\end{pmatrix}$ ($\therefore$ not entangled), but $\frac{1}{\sqrt{2}}(0 \quad 1 \quad 1 \quad 0)^\mathsf{T}$ cannot be factored ($\therefore$ entangled)

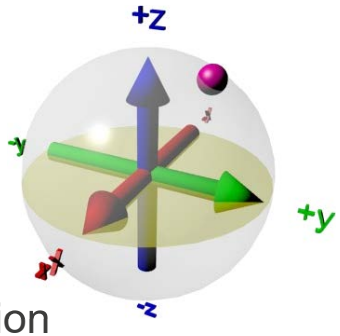
Basic Circuit-Model Concepts

- **Analogy to classical, digital circuits**



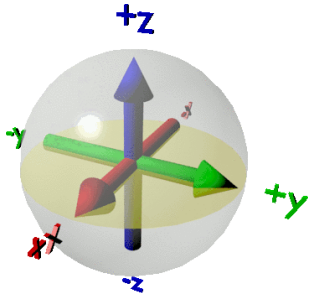


- **Differences**
  - Quantum circuits must be reversible (implication: same number of inputs and outputs for each gate and for the circuit as a whole)
  - Only combinational, not sequential, logic
- **Key point**
  - Abstract model of the operators to be applied—software not hardware
- **A qubit's state can be considered a point on the unit sphere**
- **Programmers explicitly control quantum effects**
  - Superposition: This qubit should be rotated by this amount in this direction
  - Entanglement (loosely): This qubit should conditionally rotate that qubit



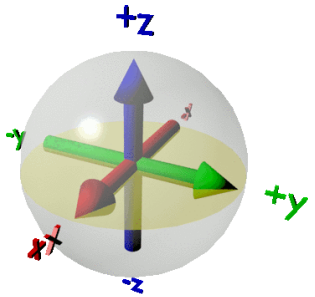
Manipulating Quantum States

- **Apply *operators* (a.k.a. *quantum gates*)**
  - Unitary matrices (corollary: all operations are reversible)
  - 2×2 for single-qubit gates, 4×4 for double-, 8×8 for triple-, etc.
- **Examples of single-qubit gates**
  - *X*, a.k.a. Pauli *x*, a.k.a. σ_x, a.k.a. NOT rotates by π radians around the *x* axis; it flips $|0\rangle \leftrightarrow |1\rangle$
  - *Y*, a.k.a. Pauli *y*, a.k.a. σ_y rotates by π radians around the *y* axis
  - *Z*, a.k.a. Pauli *z*, a.k.a. σ_z rotates by π radians around the *z* axis
  - Note that $XX = YY = ZZ = I$
- **A rotation in any direction by any amount is a gate**
  - Example: $\sqrt{\text{NOT}}$ rotates by $\pi/2$ radians around the *x* axis



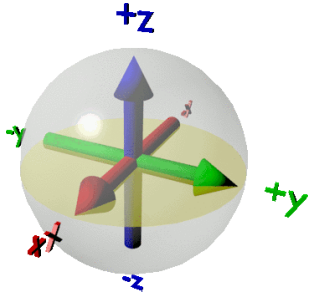
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Pauli *x* gate



$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Pauli *y* gate

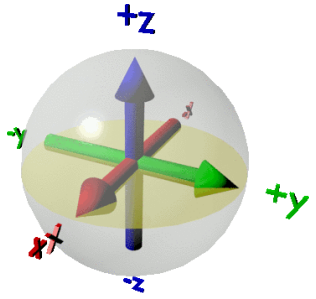


$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Pauli *z* gate

Manipulating Quantum States (cont.)

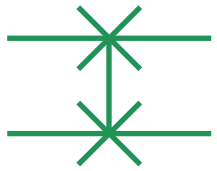
- **An important single-qubit gate**
  - *H*, a.k.a. Hadamard rotates by π radians around the diagonal pointing towards (+*x*, +*z*); it puts each of $|0\rangle$ and $|1\rangle$ into a perfect superposition of $|0\rangle$ and $|1\rangle$
    - $|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, a.k.a. $|+\rangle$
    - $|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, a.k.a. $|-\rangle$
  - Measurement of perfect superposition returns 0 and 1 with equal probability
  - *Surprise*: applying a Hadamard gate to a perfect superposition returns 0 or 1 with certainty (because $HH = I$)



Hadamard gate

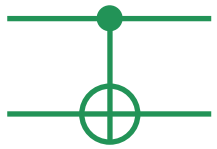
$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

- **Examples of two-qubit gates**
  - SWAP: $\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$



Swaps the values of the two qubits (i.e., maps $|ab\rangle \rightarrow |ba\rangle$)

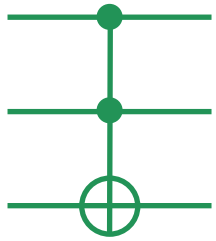
  - CNOT: $\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$



Flips the second qubit if and only if the first qubit is 1 [“**if** *a* **then** $b \leftarrow \neg b$”] (essentially an XOR: $|ab\rangle \rightarrow |a\rangle|a \oplus b\rangle$)
  - Side effect of entangling the two qubits

A Useful Three-Qubit Gate

- **Toffoli gate**
  - A.k.a. controlled-controlled-not or CCNOT
  - CCNOT: $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$



  - Flips the third qubit if and only if *both* of the first two qubits are 1
  - Maps $|abc\rangle \rightarrow |a\rangle|b\rangle|c \oplus ab\rangle$
- **Universal gate**
  - Can implement any classical Boolean function using only CCNOTs
  - **AND**: CCNOT(x, y, 0) → (x, y, $x \wedge y$)
  - **NOT**: CCNOT(1, 1, x) → (1, 1, $\neg x$)
  - **OR**: CCNOT(1, 1, CCNOT(CCNOT(1, 1, x), CCNOT(1, 1, y), 0)) → (1, 1, $\neg x$, $\neg y$, $x \vee y$)
  - **NAND**: CCNOT(x, y, 1) → (x, y, $\neg(x \wedge y)$)

Input	Output
$\|000\rangle$	$\|000\rangle$
$\|001\rangle$	$\|001\rangle$
$\|010\rangle$	$\|010\rangle$
$\|011\rangle$	$\|011\rangle$
$\|100\rangle$	$\|100\rangle$
$\|101\rangle$	$\|101\rangle$
$\|110\rangle$	$\|111\rangle$
$\|111\rangle$	$\|110\rangle$

Constructing a Gate from First Principles

- **What matrix implements a Pauli *X* (NOT) gate?**
  - We assume the standard basis, $|0\rangle \equiv \begin{pmatrix}1\\0\end{pmatrix}$ and $|1\rangle \equiv \begin{pmatrix}0\\1\end{pmatrix}$
- **Start with a truth table mapping inputs to outputs**

Input	Output
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

- **Define a corresponding operator**
  - One term per row, which maps input to output and all else to the zero vector
  - $X = |1\rangle\langle 0| + |0\rangle\langle 1|$
  - In matrix form, this would be $X = \begin{pmatrix}0\\1\end{pmatrix}(1 \quad 0) + \begin{pmatrix}1\\0\end{pmatrix}(0 \quad 1) = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$
- **Although defined using basis vectors, this works on superpositions, too**
  - Example: If $|\psi\rangle \equiv \sqrt{\frac{1}{4}}|0\rangle - \sqrt{\frac{3}{4}}|1\rangle$, then $X|\psi\rangle = -\sqrt{\frac{3}{4}}|0\rangle + \sqrt{\frac{1}{4}}|1\rangle$

Constructing a Larger Gate from First Principles

- **What operator/matrix implements a SWAP gate?**
  - This is a two-qubit gate with the semantics $|ab\rangle \rightarrow |ba\rangle$
- **The corresponding truth table is shown at right**
- **Construct an operator (same process as before but with more terms)**
  - $SWAP = |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11|$

Input	Output
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|10\rangle$
$\|10\rangle$	$\|01\rangle$
$\|11\rangle$	$\|11\rangle$

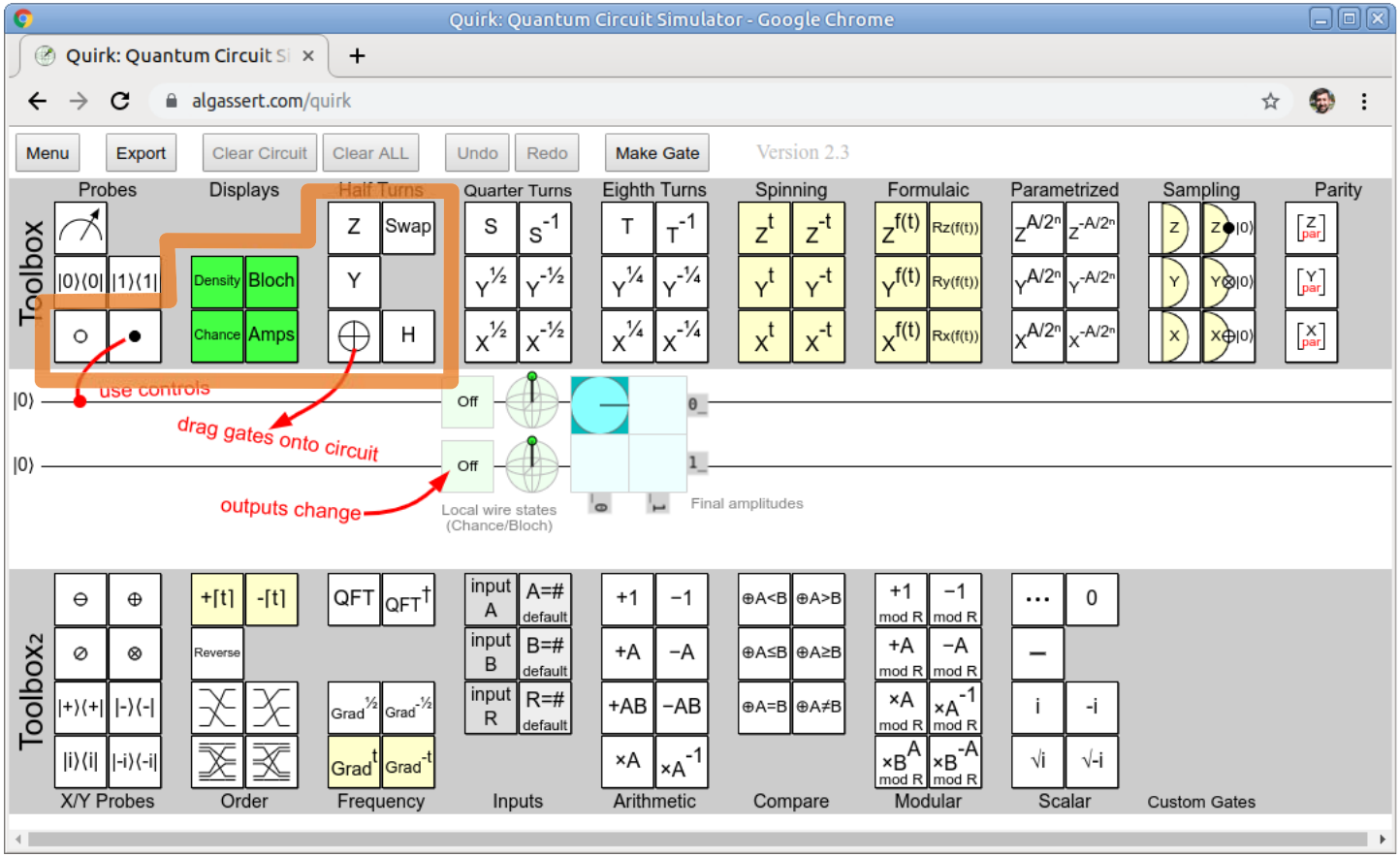
$$= \begin{pmatrix}1\\0\\0\\0\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix} + \begin{pmatrix}0\\0\\1\\0\end{pmatrix}\begin{pmatrix}0 & 1 & 0 & 0\end{pmatrix} + \begin{pmatrix}0\\1\\0\\0\end{pmatrix}\begin{pmatrix}0 & 0 & 1 & 0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}$$

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

Let's Create a Quantum Circuit

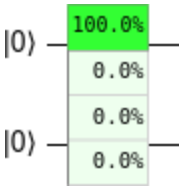
- **We'll use the Quirk gate-model simulator for this task**
  - Go to https://algassert.com/quirk and click Edit Circuit
  - Easy to use; lots of features; runs entirely within a Web browser

For now, we'll focus on just the most basic gates

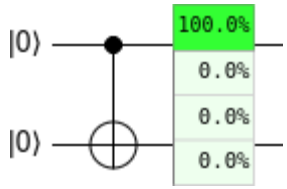


Let's Create a Quantum Circuit (cont.)

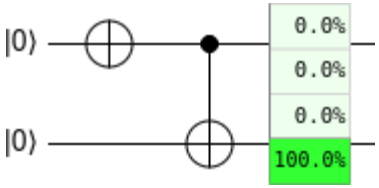
- **What state are we in initially?**
  - The $|00\rangle$ state
  - Place a Chance display on qubit 0 then extend it downwards to cover qubit 1, which shows all two-qubit probabilities



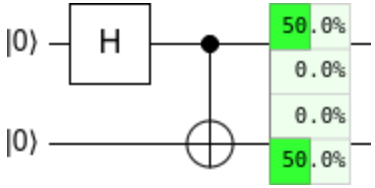
- **What if we add a CNOT from 0 to 1?**
  - So far, nothing happens ($|00\rangle \rightarrow |00\rangle$)



- **What if we put an X before the control?**
  - The state changes from $|00\rangle$ to $|11\rangle$

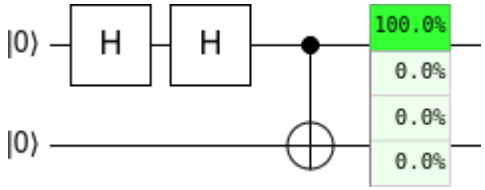


- **What if change the X to an H?**
  - We're now in the state $|00\rangle + |11\rangle$
  - Because qubit 0 is now equally $|0\rangle$ and $|1\rangle$, it both flips and doesn't flip qubit 1

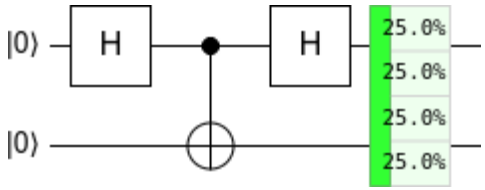


Let's Create a Quantum Circuit (cont.)

- **What if we double the H?**
  - We're back in the $|00\rangle$ state
  - H-H = I so qubit 0 is 0 and we therefore don't flip qubit 1

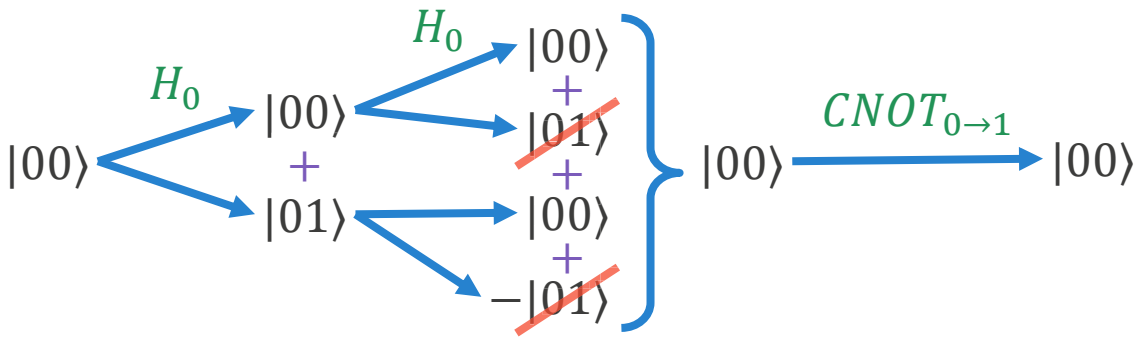


- **What if we move one of the Hs after the CNOT control?**
  - We're in the $|00\rangle + |01\rangle + |10\rangle - |11\rangle$ state



Whoa! What Just Happened?

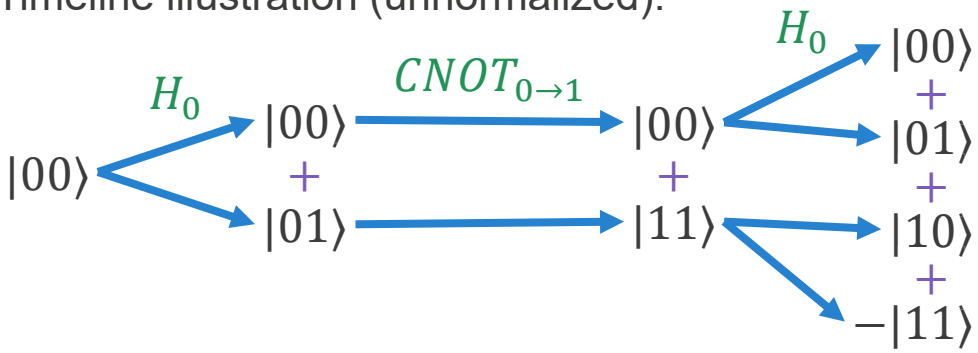
- **Why does H-H-CNOT produce such a different result from H-CNOT-H?**
  - Let's step through the two cases slowly to see what each circuit does…
- **The H-H-CNOT case**
  - Timeline illustration (unnormalized):

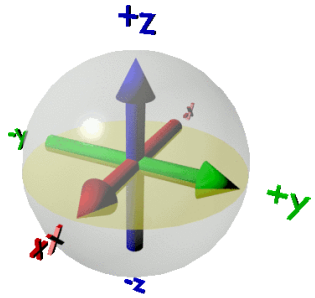


The H gate (unnormalized)

Input	Output
$\|0\rangle$	$\|+\rangle = \|0\rangle + \|1\rangle$
$\|1\rangle$	$\|-\rangle = \|0\rangle - \|1\rangle$

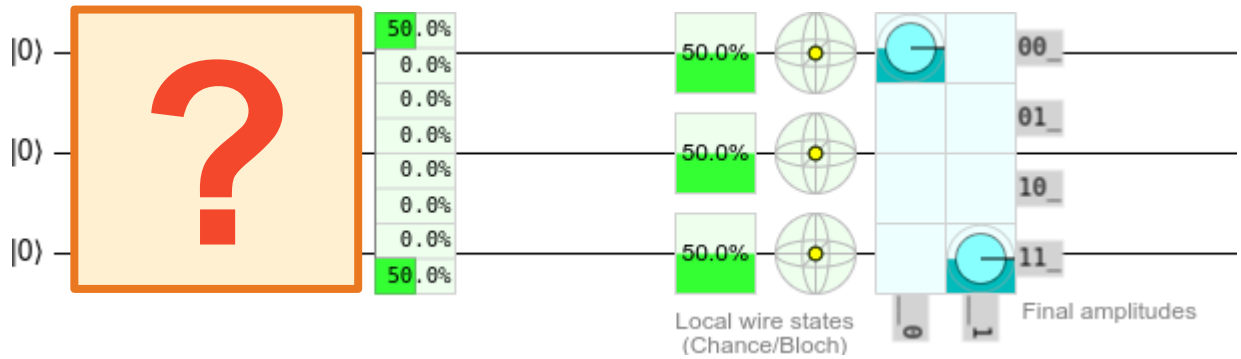
- **The H-CNOT-H case**
  - Timeline illustration (unnormalized):





Hands-On Exercise: Construct a 3-Qubit GHZ State

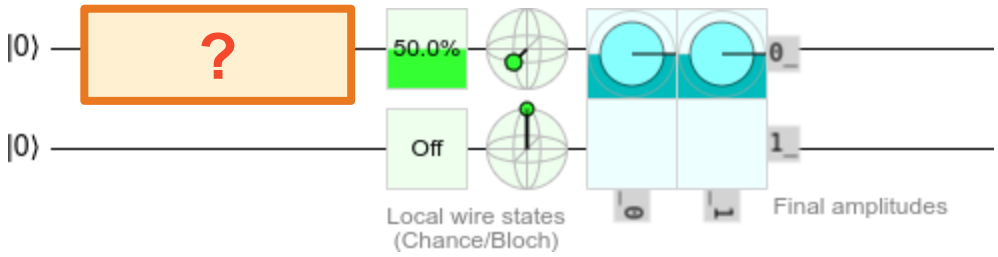
- **Greenberger–Horne–Zeilinger (GHZ) state**
  - Entangled state, equally likely to be all zeros or all ones but never anything else
- **For this exercise, we'll construct a 3-qubit GHZ state in Quirk**
  - That is, we want to create a circuit that produces $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$
  - Here's what your solution should look like (and note that we extended the Chance display to cover three qubits):



- **We'll provide hints every few minutes to help you keep making progress**

3-Qubit GHZ State: Hint #1

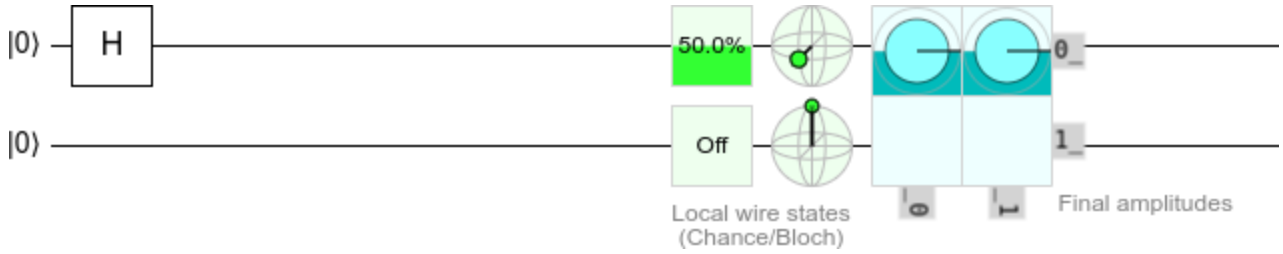
- **How would you create a 1-qubit GHZ state?**
  - That is, $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (a.k.a. $|+\rangle$), a state that's equally likely to be $|0\rangle$ or $|1\rangle$
  - What gate have we seen that does this?
- **Solution format**



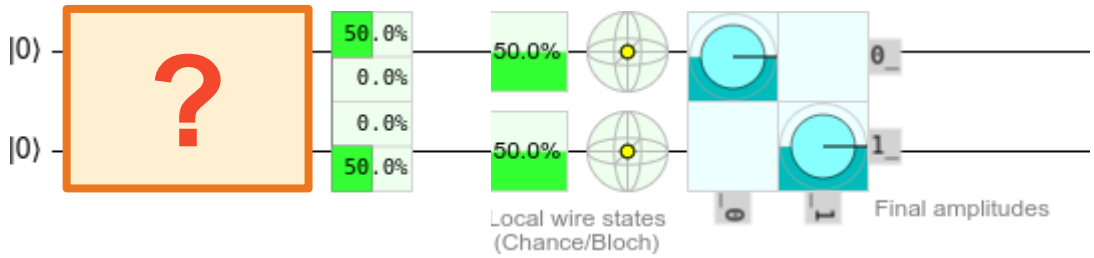
  - Quirk requires a minimum of two qubits so just leave qubit 1 alone
  - Technically, the above represents $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$

3-Qubit GHZ State: Hint #2

- **Solution to Hint #1: Creating a 1-qubit GHZ state**
  - All we need is an H gate to transform state $|00\rangle$ into state $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$



- **Hint #2: How would you create a 2-qubit GHZ state?**
  - That is, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, a state that's equally likely to be $|00\rangle$ or $|11\rangle$
  - Start from the Hint #1 state, $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$
  - How can we leave $|00\rangle$ alone but replace $|01\rangle$ with $|11\rangle$?
- **Solution format**

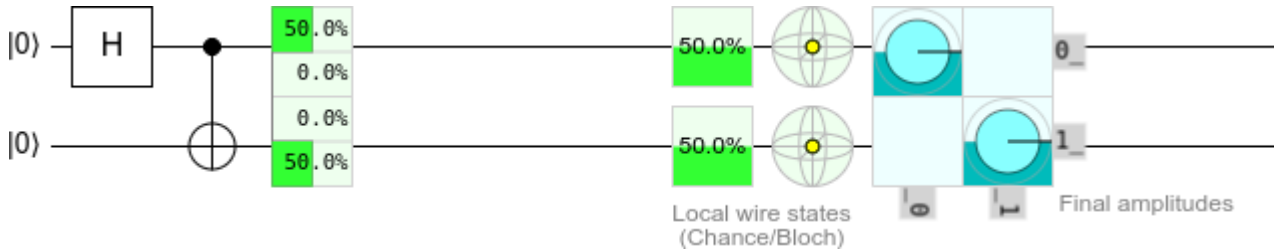


3-Qubit GHZ State: Hint #2′

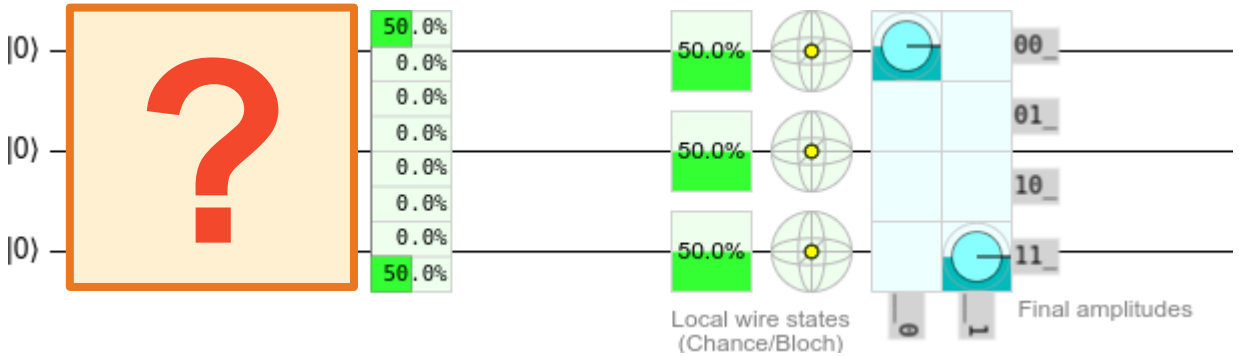
- **Hint #2: How would you create a 2-qubit GHZ state?**
  - That is, $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, a state that's equally likely to be $|00\rangle$ or $|11\rangle$
  - Start from the Hint #1 state, $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$
  - How can we leave $|00\rangle$ alone but replace $|01\rangle$ with $|11\rangle$?
- **Hint #2′: What single 2-qubit gate performs the preceding mapping?**
  - Given $|ab\rangle$, negate a if and only if b is 1

3-Qubit GHZ State: Hint #3

- **Solution to Hint #2: Creating a 2-qubit GHZ state**
  - A CNOT gate performs the requisite mapping from $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$ to $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
  - "If qubit 0 is 1, flip qubit 1" (from 0 to 1 in this case)

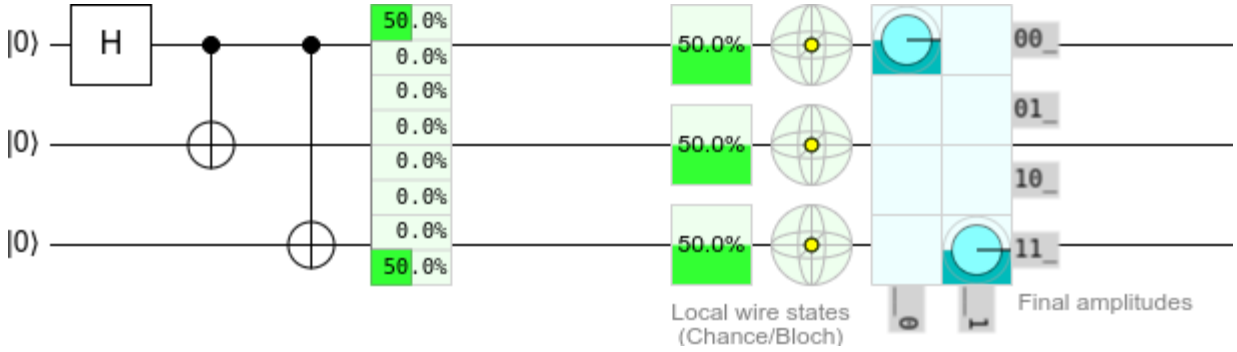


- **Hint #3: How would you create a 3-qubit GHZ state?**
  - Extend the above to 3 qubits: Given $\frac{1}{\sqrt{2}}(|000\rangle + |011\rangle)$, produce $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$
- **Solution format**



3-Qubit GHZ State: Solution

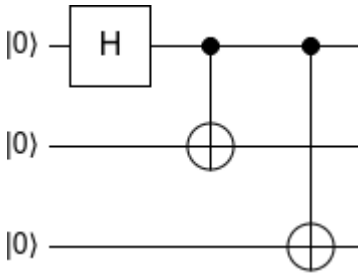
- **Solution to Hint #3: Creating a 3-qubit GHZ state**
  - We simply repeat what we did for Hint #2
  - A CNOT from qubit 0 to qubit 2 implements “If qubit 0 is 1, flip qubit 2”
  - Maps $\frac{1}{\sqrt{2}}(|000\rangle + |011\rangle)$ to $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$



- **Continuing the pattern**
  - For a 4-qubit GHZ state, add a CNOT from qubit 0 to qubit 3
  - For a 5-qubit GHZ state, add a CNOT from qubit 0 to qubit 4
  - For a 6-qubit GHZ state, add a CNOT from qubit 0 to qubit 5

Comparing Circuit and Matrix Formulations

- **Note how much easier it is to specify a quantum circuit gate-by-gate than to specify the complete unitary matrix to which it corresponds:**

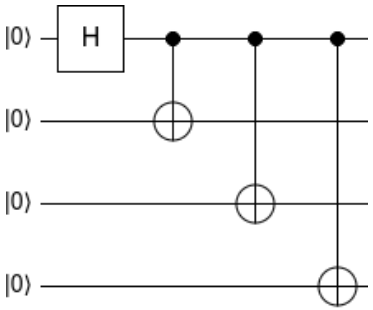


vs.

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Comparing Circuit and Matrix Formulations (cont.)

- **Let's extend the 3-qubit GHZ state to a 4-qubit GHZ state**
  - Add one more CNOT to the circuit or double each matrix dimension



vs.

$$\frac{1}{\sqrt{2}}\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

- **Very quickly grows out of hand**
  - A 10-qubit GHZ state could be expressed with either an H and 10 CNOTs or a million-element unitary matrix

Basic Quantum Algorithms

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

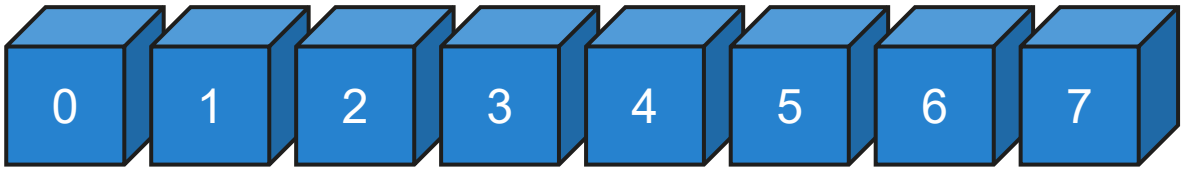
Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

Grover's Algorithm

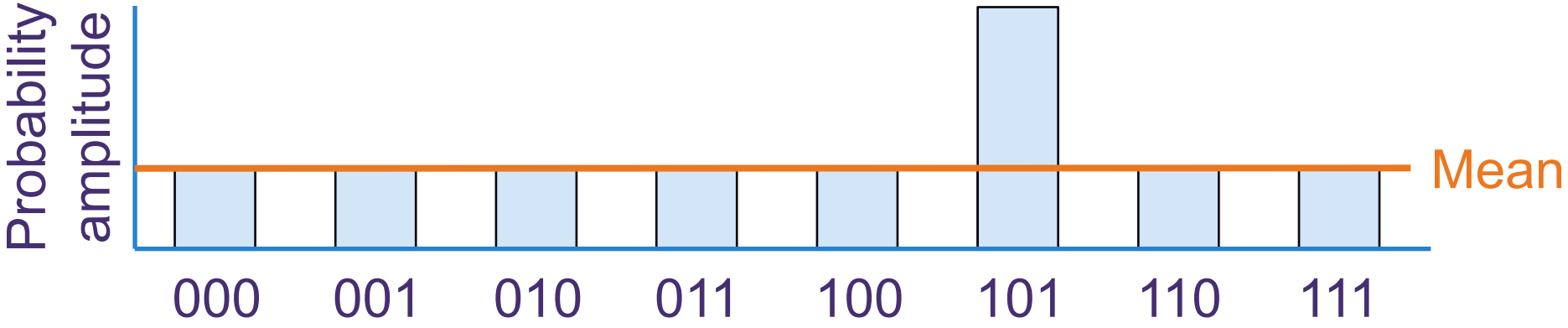
- **Which box contains the prize?**



  - Classically, must open all 8 boxes in the worst case
- **Let's see how we can use quantum effects to do better than that…**
- **Given**
  - A power-of-two number of boxes
  - A guarantee that exactly one box contains the prize
  - An operator U_ω that, given a box number $|x\rangle$, negates the probability amplitude iff the box contains the prize (i.e., $U_\omega|x\rangle = -|x\rangle$ for $x = \omega$ and $U_\omega|x\rangle = |x\rangle$ for $x \neq \omega$)
- **Define the *Grover diffusion operator* as follows**
  - $|s\rangle \equiv \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ (i.e., the equal superposition of all states)
  - $U_s \equiv 2|s\rangle\langle s| - I$ (the Grover diffusion operator)

Grover's Algorithm (cont.)

- **The basic algorithm is fairly straightforward to apply:**
  - Put each of the *n* qubits in a superposition of $|0\rangle$ and $|1\rangle$
  - For $\left\lfloor \frac{\pi}{4}\sqrt{2^n} \right\rfloor$ iterations
    - Apply U_ω to the state
    - Apply U_s to the state
- **How does that work?**
  - Gradually shifts the probability amplitude to state $|\omega\rangle$ from all the other states
  - When we measure, we'll get a result of $|\omega\rangle$ with near certainty

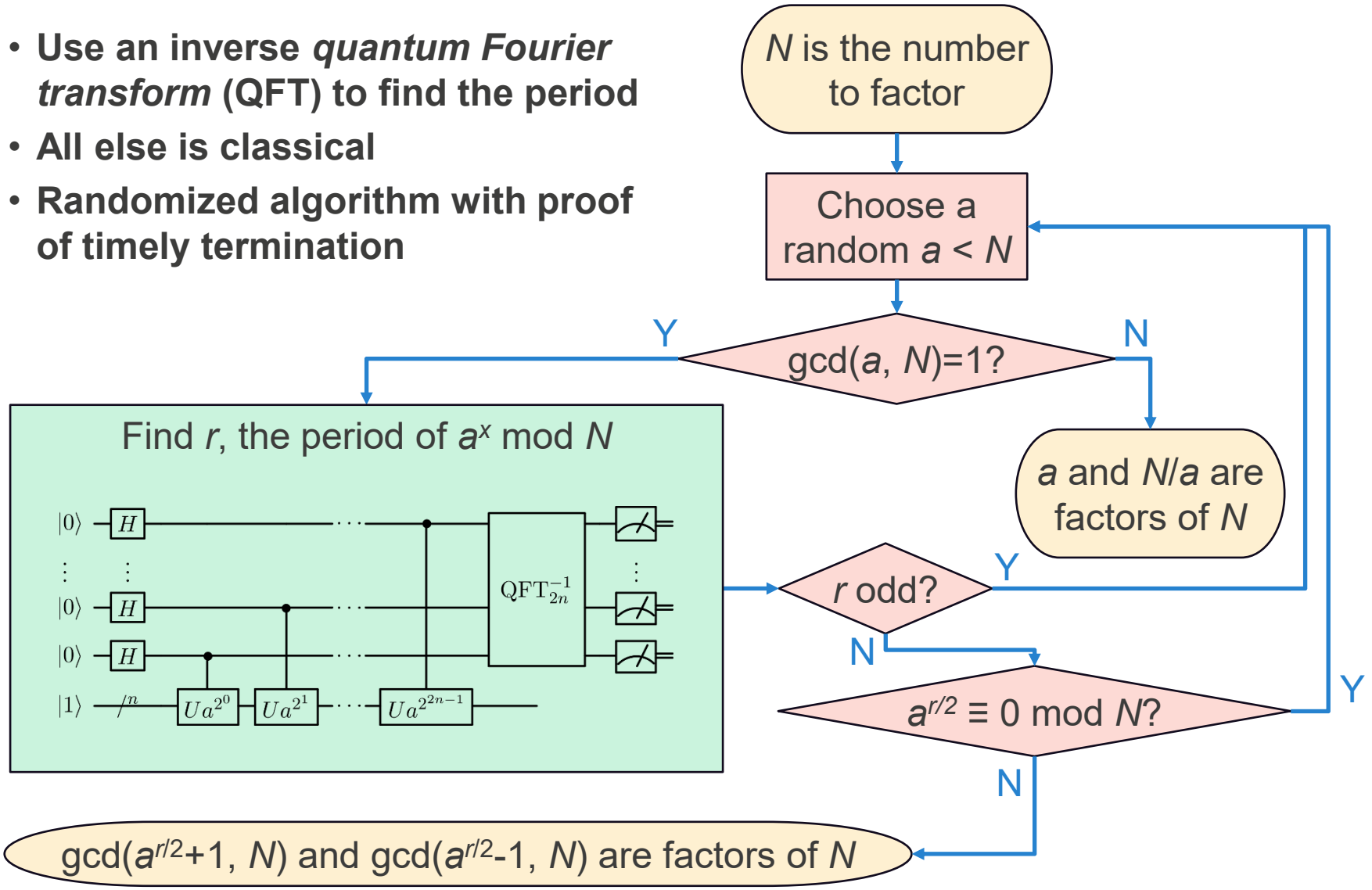


Shor's Algorithm

- **Factor 1,274,093,332,123,426,680,869 into a product of two primes**
  - Okay, it's 135,763,451,261×9,384,656,329
- **Observations**
  - Given that N is the product of two primes, p and q
  - Given some a that is not divisible by either p or q
  - Then the sequence $\{a^1 \bmod N, a^2 \bmod N, a^3 \bmod N, a^4 \bmod N, a^5 \bmod N, \ldots\}$ will repeat every r elements (the sequence's *period*)
  - As Euler discovered (ca. 1760), r always divides $(p-1)\,(q-1)$
- **Example**
  - Let a be 2 and N be 15 (=3×5)
  - Then $a^x \bmod N = \{2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1 \ldots\}$ so r is 4
  - Lo and behold, 4 divides (3−1) (5−1)=8
- **Approach**
  - Once we know the period, r, it's not too hard to find N's prime factors, p and q
  - Unfortunately, finding r is extremely time-consuming…for a classical computer

Shor's Algorithm (cont.)

- **Use an inverse *quantum Fourier transform* (QFT) to find the period**
- **All else is classical**
- **Randomized algorithm with proof of timely termination**



Further Quantum Algorithms and Tools

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

Target for NISQ Evaluation: Quantum Optimization Heuristics

- **Instances of combinatorial optimization problems**
  - Current approach: classical heuristics algorithms
  - NISQ hardware provides means to evaluate quantum heuristic algorithms

One strategy: Try the simplest algorithm that might work!

- **Quantum heuristics**
  - Combine cost-function-based operator with a mixing operator
  - AQO, QA, QAOA
  - Other ideas welcome!

- **Evaluation techniques**
  - Analytic, numerical, experimenting on NISQ hardware

Target for NISQ Evaluation: Quantum Optimization Heuristics

- **Diverse optimization goals**
  - Exact optimization with guarantees
  - Approximate opt. with guarantees
  - Good heuristic, without guarantees
  - Fair sampling; portfolio sampling
- **Sampling goals, e.g. for machine learning (ML)**
  - Sampling thermal distribution corresponding to cost function (sampling from Boltzmann distributions used in ML)

Quantum Optimization Algorithms: AQO, QA, QAOA

- **Common elements: Given cost function C(z),**
- **Phase separation operator based on the cost function,**
  - Usually based on $H_P = -\Sigma C(z)|z\rangle\langle z|$, often including additional "penalty terms" to enforce constraints
- **Driver/Mixing operator**
  - Most frequently $H_M = \sum_j Xj$, though we will shortly see other mixers

AQO

- Evolution under

$$\boldsymbol{H(t) = a(t)H_P + b(t)H_M}$$

- Slowly enough to stay in the ground subspace

QA

- Evolution under

$$\boldsymbol{H(t) = a(t)H_P + b(t)H_M}$$

- Many quick runs, thermal effect contribute

QAOA

- Alternate application of H_P and H_M
- For $\boldsymbol{p}$ alterations, the parameters are $\boldsymbol{2p}$ times/angles $\boldsymbol{\gamma_1, \beta_1, \ldots \gamma_P, \beta_p}$

Quantum Alternating Operator Ansatz

- **Advantages**
  - Supports more general mixing operators, providing massive improvements in implementability
  - Incorporates hard constraints into mixer instead of as a penalty term; algorithm explores only feasible subspace, often exponentially smaller, so more efficient search
  - Reworked QAOA acronym to support applications to exact optimization and sampling as well as approximate optimization
- **Many problems can be mapped to extended QAOA formalism**
  - Initial paper focused on scheduling and network problems

S. Hadfield et al., **From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz**, Algorithms 12 (2), 34 2019, arXiv:1709.03489

Summary and Open Questions

Unclear as of yet as to whether QA or QAOA provides a quantum advantage beyond a few examples

- True for ***any*** NISQ quantum optimization algorithm!

Parameter setting is challenging

- Active area of research
- relation between parameter setting in QAOA and annealing schedule choice in quantum annealing
- Only requires ***satisficing***, not optimizing

Exploration of variants of QAOA and QA may be promising

Empirical evaluation of QAOA as a quantum heuristic critical for understanding its potential impact

Tie between quantum hardware and quantum algorithms research

- Role of special purpose quantum hardware
- Possibility of quantum hardware between quantum annealers and gate model, pulsing global Hamiltonians

Connecting QA schedules and QAOA parameter setting

Yang, Rahmani, Shabani, H Neven, C Chamon, *Optimizing variational quantum algorithms using pontryagin's minimum principle*, PRX 2017

- Pontryagin's minimum principle implies optimal evolution schedules must be bang-bang, up to some caveats

Zhou, Wang, Choi, Pichler, Lukin, *Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices*, arXiv:1812.01041

- Learned optimal parameters
- Identified regular subfamily of optimal parameters, resembling digitized smooth evolution
  - For easy problems, resembled adiabatic schedules
  - For hard problems, resembled diabatic schedules

Mbeng, Fazio, Santoro, *Quantum Annealing: a journey through Digitalization, Control, and hybrid Quantum Variational schemes*, arXiv:1906.08948

- Connects adiabatic adiabatic schedules with optimal QAOA parameters for the easy problem of MaxCut on a Ring

Brady, Baldwin, Bapat, Kharkov, V. Gorshkov, *Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems*, arXiv:2003.08952

- generically, for a fixed amount of time, optimal procedure has bang-bang structure of QAOA at the beginning and end, but a smooth annealing structure in between

LT Brady, L Kocia, P Bienias, A Bapat, Y Kharkov, AV Gorshkov, *Behavior of analog quantum algorithms*, arXiv:2107.01218

- optimal procedure approaches a smooth adiabatic procedure but with a superposed oscillatory pattern
- QAOA emulates this optimal procedure
- new algorithm that better approximates the optimal protocol

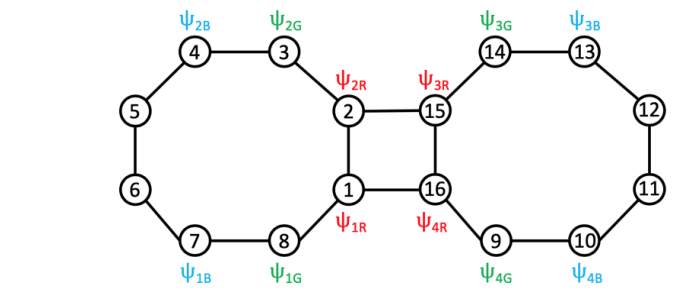
Qubit Routing on NISQ Processors

Compilation of an algorithms to a NISQ processor requires

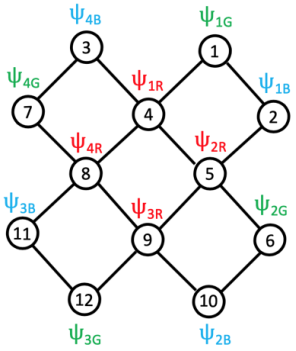
- Decomposition into native gates
- Qubit routing

Qubit routing moves qubit states to locations where the required gates can act on them

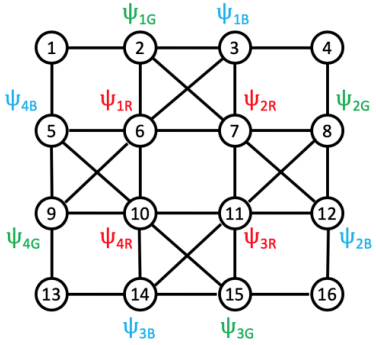
- Can be done by inserting SWAPs into a circuit composed of native



(a) Rigetti Aspen



(b) Google Bristlecone

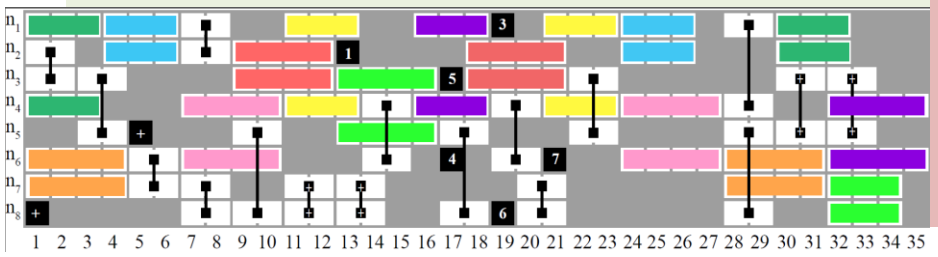


(c) IBM Tokyo

From Minh Do, Zhihui Wang, Bryan O'Gorman, Davide Venturelli, Eleanor Rieffel, Jeremy Frank, ***Planning for Compilation of a Quantum Algorithm for Graph Coloring****, ECAI 2020, arXiv:2002.10917*

Temporal Planning for Qubit Routing

- **Qubit routing can be phrased as a temporal planning problem**
  - minimize *makespan*
- **Can incorporate**
  - nearest-neighbor h/w constraints
  - varying quantum gate times
  - crosstalk
- **Initial experiments focused on**
  - QAOA circuits for Maxcut because of their high number of commuting gates
  - Rigetti hardware proposal with varying gates between neighboring qubits



- **Mapped circuit compilation problem to a temporal planning problem, compared state-of-the-art temporal planners**
- **Demonstrated temporal planning is a viable approach to circuit compilation**
- **More recently, combined temporal planning with constrained programming**
- **Expressive framework can incorporate further hardware requirements, incl. noise tradeoffs, as we learn them**

D. Venturelli et al., Compiling quantum circuits to realistic hardware architectures using temporal planners, *Quantum Science and Technology* (2018)

Classical HPC Simulation of Quantum Circuits

Advanced the state-of-the-art

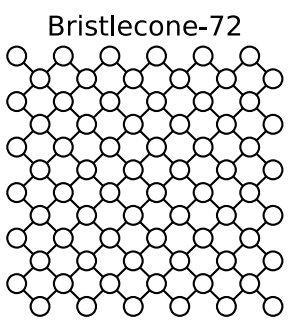
- simulates **larger quantum circuits** than previous approaches
- **judicious use of cuts** within a tensor network contraction
- **HPC memory tricks** and trade-offs
- can flexibly **incorporate fidelity** goal

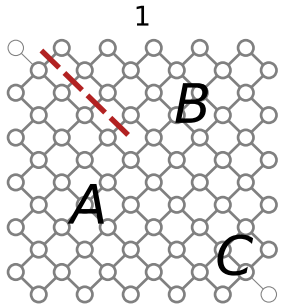
Largest computation run on NASA HPC clusters

- 60-qubit subgraph, depth 1+32+1
- 116,611 processes on 13,059 nodes, peak of 20 PFLOPS, 64% of max
- across Pleiades, Electra, Hyperwall

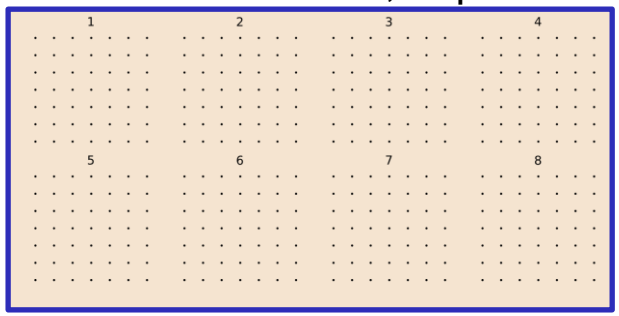
Applications

- quantum supremacy experiments
- benchmark emerging quantum hardware
- empirically explore quantum algorithms





Computed exact amplitudes for 72 qubit Bristlecone random circuit, depth 1+32+1



Villalonga et al., ***A flexible high-performance simulator for the verification and benchmarking of quantum circuits implemented on real hardware***. arXiv:1811.09599
Villalonga et al., ***Establishing the Quantum Supremacy Frontier with a 281 Pflop/s Simulation***, arXiv:1905.00444

Open source qFlex code:
https://github.com/ngnrsaa/qflex

HybridQ: A Hybrid Quantum Simulator for Large Scale Simulations

Hardware agnostic quantum simulator, designed to simulate large scale quantum circuits.

Can run tensor contraction simulations, direct evolution simulation and Clifford+T simulations using the same syntax

Features:

- Fully compatible with Python (3.8+)
- Low-level optimization achieved by using C++ and Just-In-Time (JIT) compilation with JAX and Numba,
- It can run seamlessly on CPU/GPU and TPU, either on single or multiple nodes (MPI) for large scale simulations, using the exact same syntax
- User-friendly interface with an advanced language to describe circuits and gates, including tools to manipulate/simplify circuits.

Recent Improvements:

- Commutations rules are used to simplify circuits (useful for QAOA)
- Expansion of density matrices as superpositions of Pauli strings accepts arbitrary non-Clifford gates,
- Open-source (soon!) project with continuous-integration, multiple tests and easy installation using either pip or conda

Open source code available at https://github.com/nasa/HybridQ

S. Mandrà, J. Marshall, E. G. Rieffel, R. Biswas, HybridQ: A Hybrid Simulator for Quantum Circuits, arXiv:2111.06868

Concluding Remarks

Agenda

- **Part I: Quantum-computing fundamentals**
  - High-level motivation, history, and status
  - Qubits, multi-qubit states, and quantum measurement
  - Review of notation
  - Quantum gates and quantum circuits

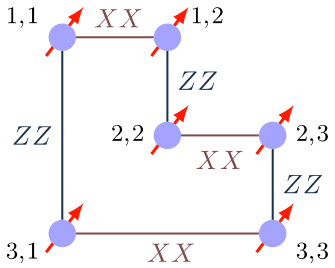
Break

- **Part II: Circuit-model quantum computing**
  - Quantum gates and quantum circuits (cont.)
  - Basic quantum algorithms
  - Further quantum algorithms and tools
  - Concluding remarks

Adjourn

Quantum Error Mitigation

- **Error suppression: Inhibits transitions out of the ground subspace**
- **Error correction: Counteracts transitions that have happened**



- **Quantum error correction initially thought impossible!**
  - No cloning principle: an unknown quantum state cannot be copied reliably without destroying the original
- **Quantum information theory was just too interesting**
  - Steane and Shor & Calderbank saw a way to finesse what had seemed insurmountable barriers to quantum error correction
- **Now quantum error correction is one of the most developed areas**
  - beautiful, almost magical, effects!
  - uses properties of quantum measurement and entanglement to its advantage
- **Stabilizer code formulation most common**
- **Subsystem codes; Dynamical Logical Qubits; LDPC codes; …**

Fault Tolerance

- **Error suppression and correction mechanisms cannot be done perfectly**
- **Fault tolerance: Ensures error suppression/correction do not introduce more problems than they solve**

- **Imprecise implementation of mechanisms may cause errors Even accurate implementation may magnify errors**
  - can take correctable errors to uncorrectable ones

- **Threshold theorems: There exists an error rate threshold below which indefinitely long quantum computations can be carried out robustly**

- In the gate model, a number of different threshold theorems are known. Specific theorems involve precise statements of error model, precision of implementation, resource quantification, distance measure
- How to establish a threshold theorem for adiabatic quantum computing remains a major open question

Measurement-Based Quantum Computing (MBQC)

- **High-level description of "one-way" measurement-based quantum computation**
  - Start in a highly entangled state that serves as the quantum resource
    - Cluster states, graph states, …
  - Make series of single-qubit measurements that can depend on previous measurement results
  - Interpret the results of the measurements to obtain a final answer
- **Properties**
  - Computational power equivalent to standard quantum computation
  - Separation between classical and quantum aspects of the computation
  - Entanglement decreases; also called one-way quantum computing
- **Resource states for MBQC**
  - Some states too entangled to serve as a resource!
  - Classically hard to sample from output distributions of non-adaptive MBQC!

Status of Quantum Algorithms

- **Anything a classical computer can do, a quantum computer can do**
- **Provable quantum advantage known for a few dozen quantum algorithms**
- **Data from Quantum Algorithms Zoo: speed up over classical**
  - Exponential: 2
  - Superpolynomial: 29
  - Polynomial: 28
  - Constant: 1
  - Varies: 4
  - Total: 64
  - https://quantumalgorithmzoo.org/
- **Rapidly expanding opportunity for empirical testing on emerging quantum hardware**

Conjecture: Quantum heuristics will significantly broaden of applications of quantum computing

What is the chance that the only cases in which quantum computing provides a speed up is in cases when we can prove it does?

Impact of Quantum Information Processing Viewpoint on Classical Computer Science

- **Analogies**
  - Complex analysis enables computation of real integrals
  - Probabilistic algorithms inform analysis of deterministic algorithms
- **Quantum computational security reductions for purely classical encryption schemes**
  - Regev's lattice-based encryption scheme
  - One of the security reductions for Gentry's fully homomorphic encryption scheme
- **Improved classical simulations of quantum systems**

- **Insights into classical complexity theory**
  - Aaronson found a short, almost trivial, proof of a property of the complexity class **PP** by showing that it is the same as the quantum complexity class **PostBQP**
- **Drucker & de Wolf (2009) survey "Quantum proofs for classical theorems"**

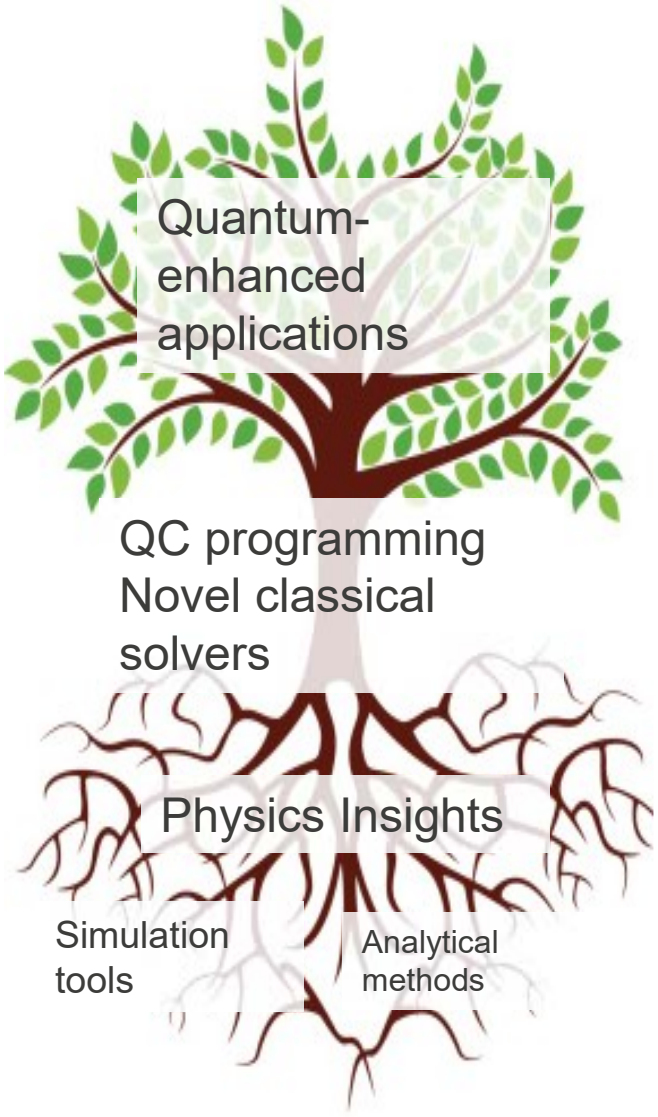
A Historical Perspective

NASA Ames director Hans Mark brought Illiac IV to NASA Ames in 1972

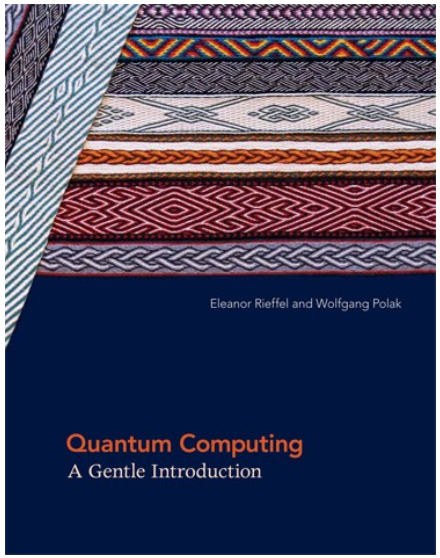
- **Illiac IV—first massively parallel computer**
  - 64 64-bit FPUs and a single CPU
  - 50 MFLOP peak, fastest computer at the time
- **Finding good problems and algorithms was challenging**
- **Questions at the time**
  - How broad will the applications be of massively parallel computing?
  - Will computers ever be able to compete with wind tunnels?

Take Away Points

- **Next year will be even more exciting!**
  - Emerging quantum hardware performing computations beyond the reach of even the largest supercomputers
- **Many open questions remain:**
  - When will scalable quantum computers be built, and how?
    - How quickly can special purpose quantum computing devices be built?
  - How broad will the impact of quantum computation be? What will the ultimate impact of quantum heuristics be?
  - How best to harness quantum effects for computational purposes?
- **Deep connection between physics and computer science**
  - How fast does nature let us compute?



Further Reading



Eleanor Rieffel and Wolfgang Polak
Quantum Computing: A Gentle Introduction
MIT Press, March 2011

And references therein

Overviews of NASA QuAIL team work

Eleanor G. Rieffel, Stuart Hadfield, Tad Hogg, Salvatore Mandrà, Jeffrey Marshall, Gianni Mossi, Bryan O'Gorman, Eugeniu Plamadeala, Norm M. Tubman, Davide Venturelli, Walter Vinci, Zhihui Wang, Max Wilson, Filip Wudarski, Rupak Biswas, ***From Ansätze to Z-gates: a NASA View of Quantum Computing***, arXiv:1905.02860

Rupak Biswas, Zhang Jiang, Kostya Kechezhi, Sergey Knysh, Salvatore Mandrà, Bryan O'Gorman, Alejandro Perdomo-Ortiz, Andre Petukhov, John Realpe-Gómez, Eleanor Rieffel, Davide Venturelli, Fedir Vasko, Zhihui Wang, ***A NASA Perspective on Quantum Computing: Opportunities and Challenges***, arXiv:1704.04836

Additional Resources

- **Free access to *physical* quantum processors**
  - IBM Quantum Experience (circuit model): https://www.ibm.com/quantum-computing/
  - D-Wave Leap (annealing model): https://cloud.dwavesys.com/leap
- **Additional software for high-level programming of D-Wave systems**
  - Prolog: QA Prolog (https://github.com/lanl/QA-Prolog)
  - C: C to D-Wave (https://github.com/lanl/c2dwave)
  - Verilog: edif2qmasm (https://github.com/lanl/edif2qmasm)
  - Macro assembly language: QMASM (https://github.com/lanl/qmasm)
  - QUBO/Ising Google Sheet (https://tinyurl.com/y6wkkkm3)—use *File→Make a copy* to store an editable version in Google Drive or *File→Download as* to save locally
- **HPC quantum-circuit simulator**
  - qFlex (https://github.com/ngnrsaa/qflex)
- **Student internships available**
  - NASA QuAIL at NASA Ames Research Center (https://ti.arc.nasa.gov/tech/dash/groups/quail/)
  - LANL Quantum Computing Summer School Fellowship: https://quantumcomputing.lanl.gov/

Funding Acknowledgements

NASA QuAIL

LANL