THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Quantum computing (QC)

Overview

September 2018

Dr. Sunil Dixit
Technical Fellow

Why is QC Important?

NORTHROP GRUMMAN

"Let's make no mistake: the competition for a quantum computer is the new arms race. The competition to create the first large-scale quantum computer is heating up. The country that develops one first will have the ability to cripple militaries and topple the global economy...

...To deter such activity, and to ensure our security, the United States must win this new race to the quantum-computer revolution."

National Review (May 2017)

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

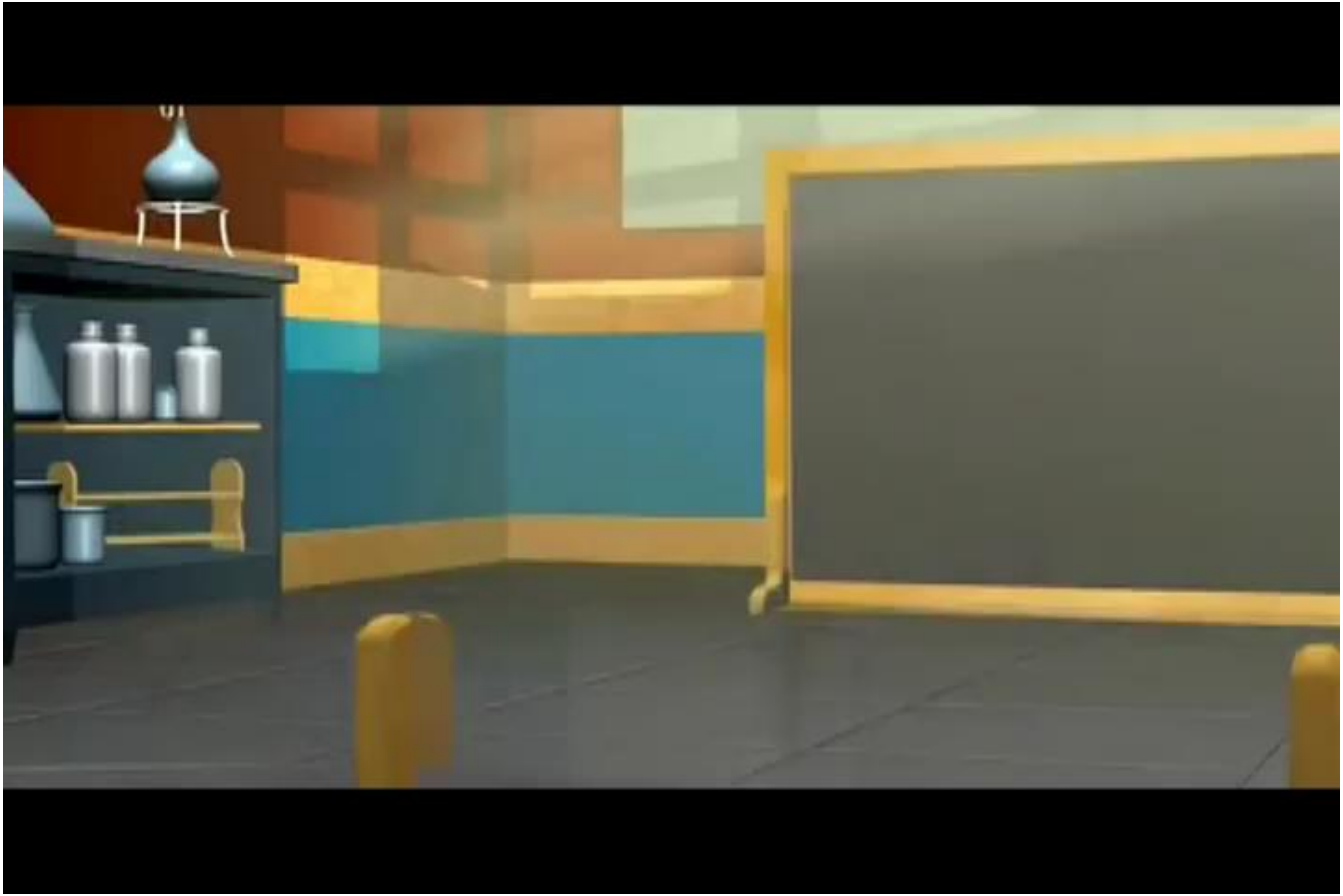
Classical Realm

Classical Physics Assumptions

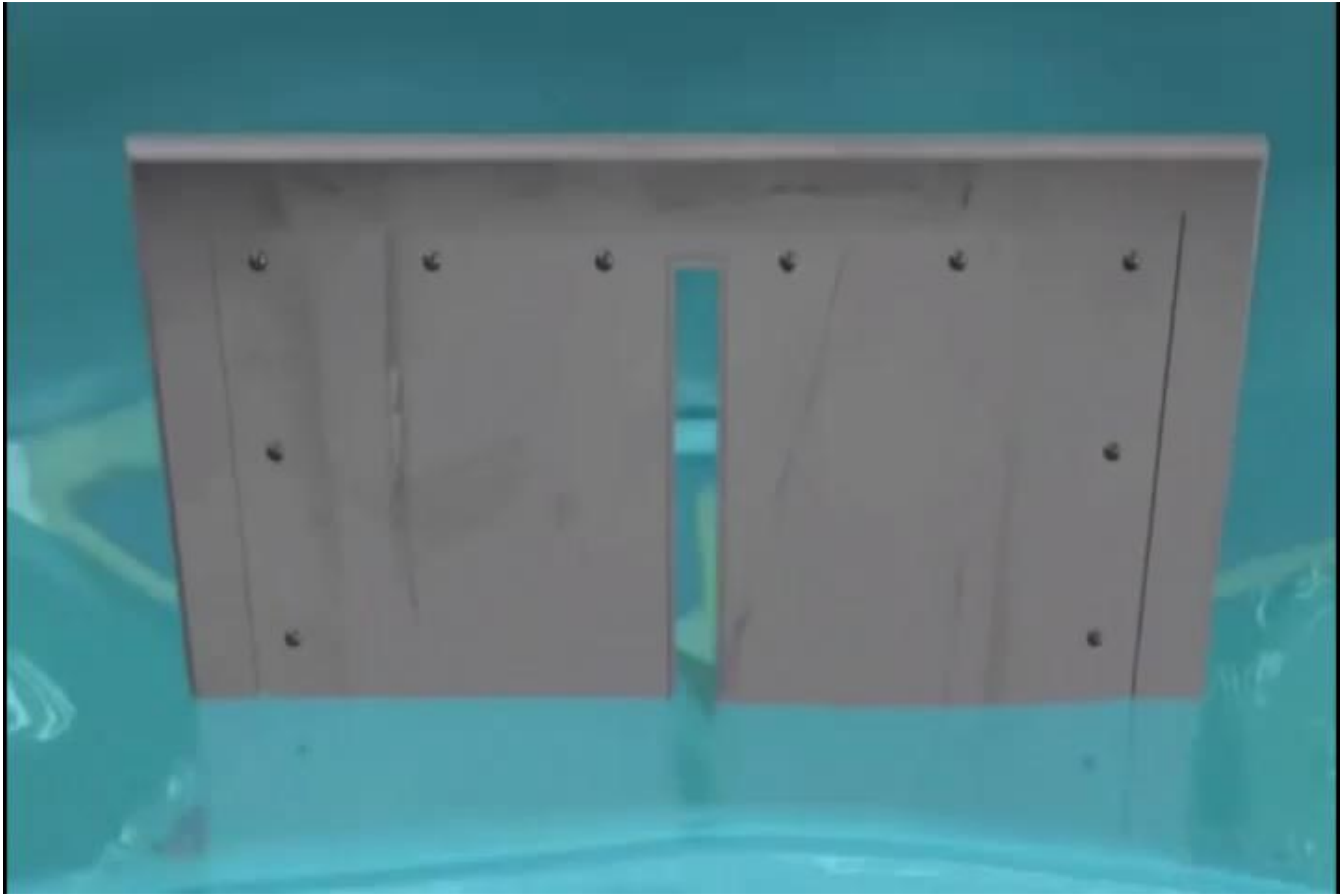
- The universe is a giant machine
- All nonuniform motion and action have cause
  - Uniform motion does not have cause (principle of inertia)
- If the state of motion is known now then all past and future states are accurately predictable because the universe is predictable
- Light is a wave described completely by Maxwell’s electromagnetic equations
- Waves and particles are distinct
- A measurement can be accurately made and errors corrected caused by the measurement tool

Single Slit – Classical Marbles

Double Slits – Classical Marbles



Single Slit – Classical Waves

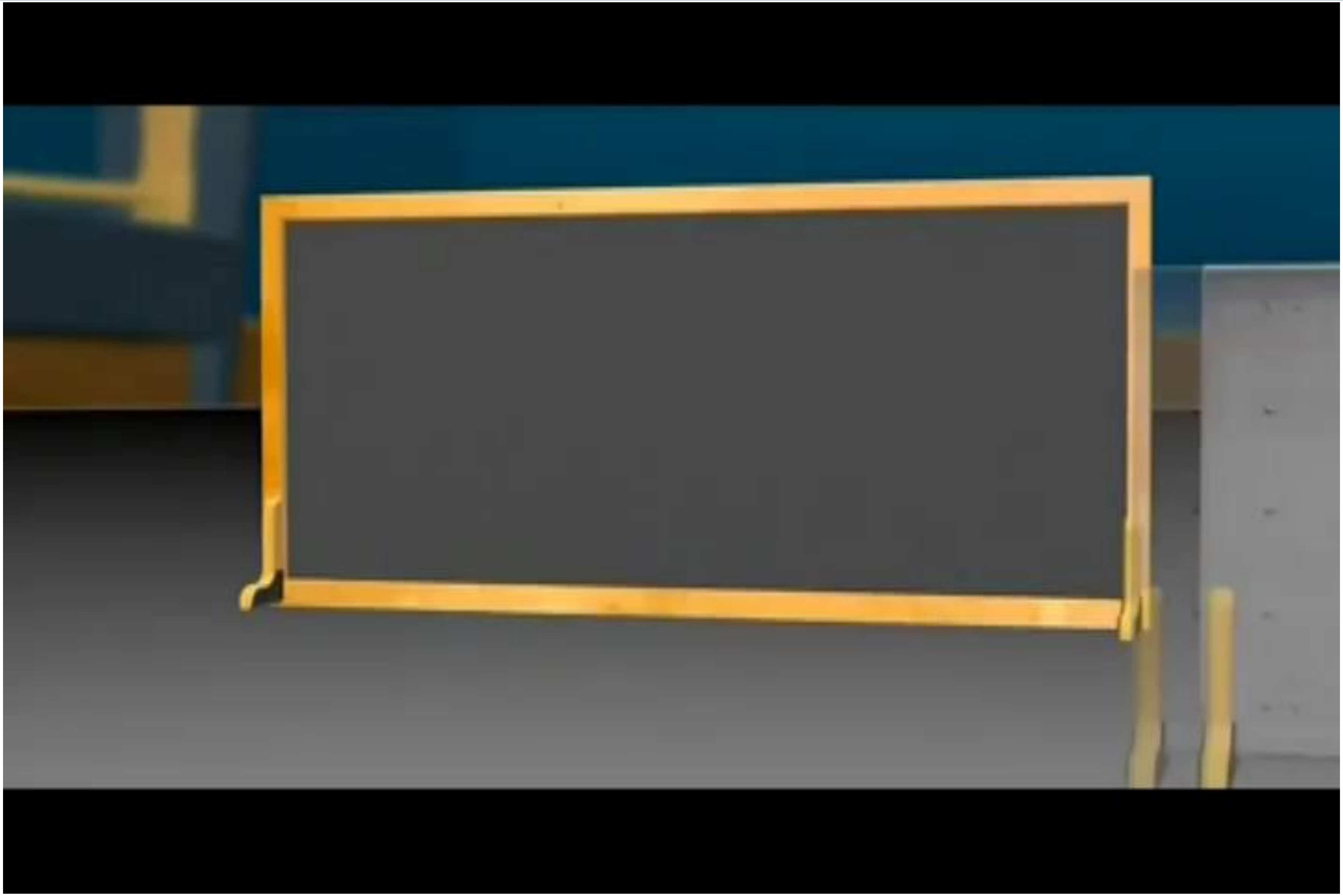


Double Slit – Classical Waves

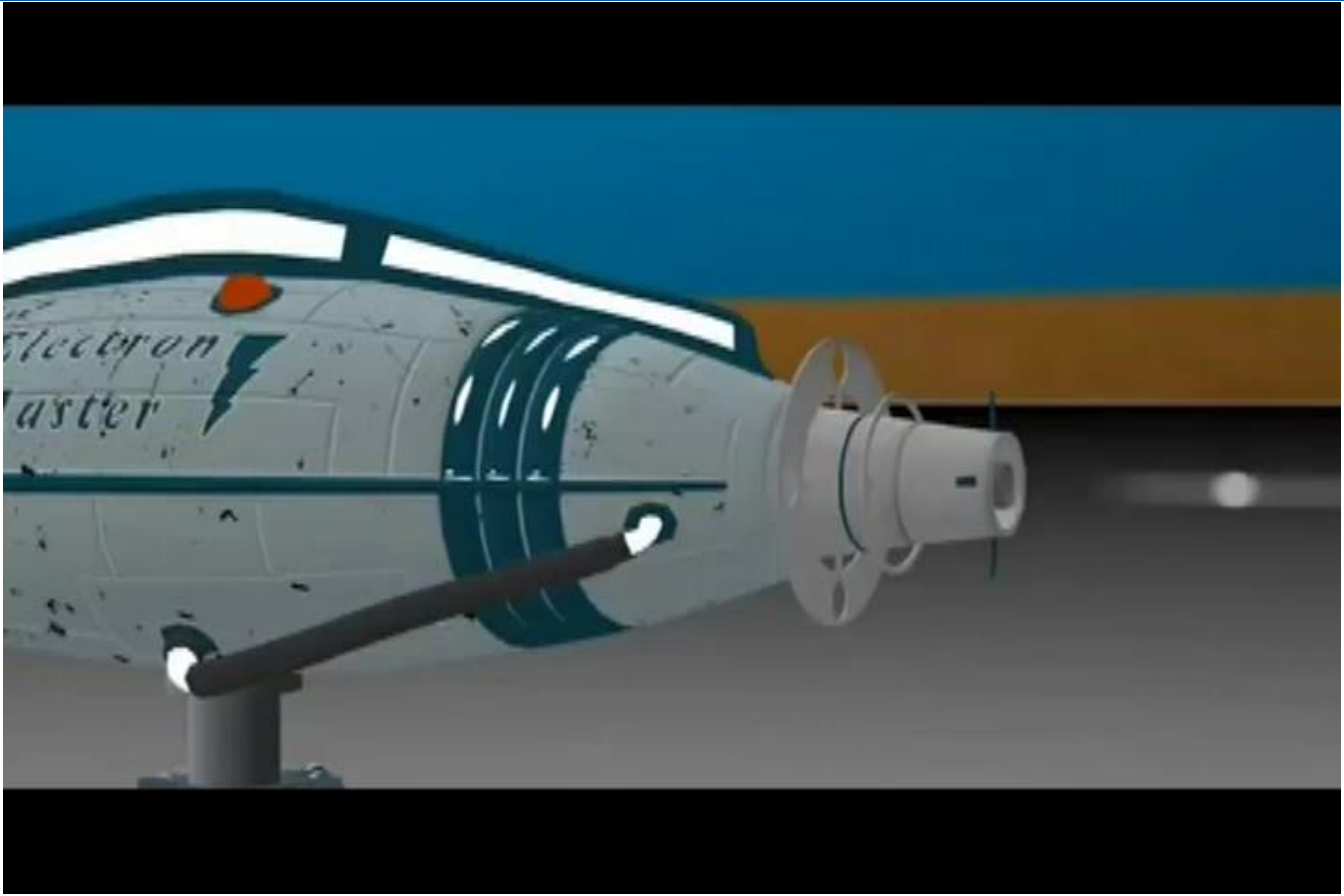
Quantum Realm

Single Slit – Quantum Electrons

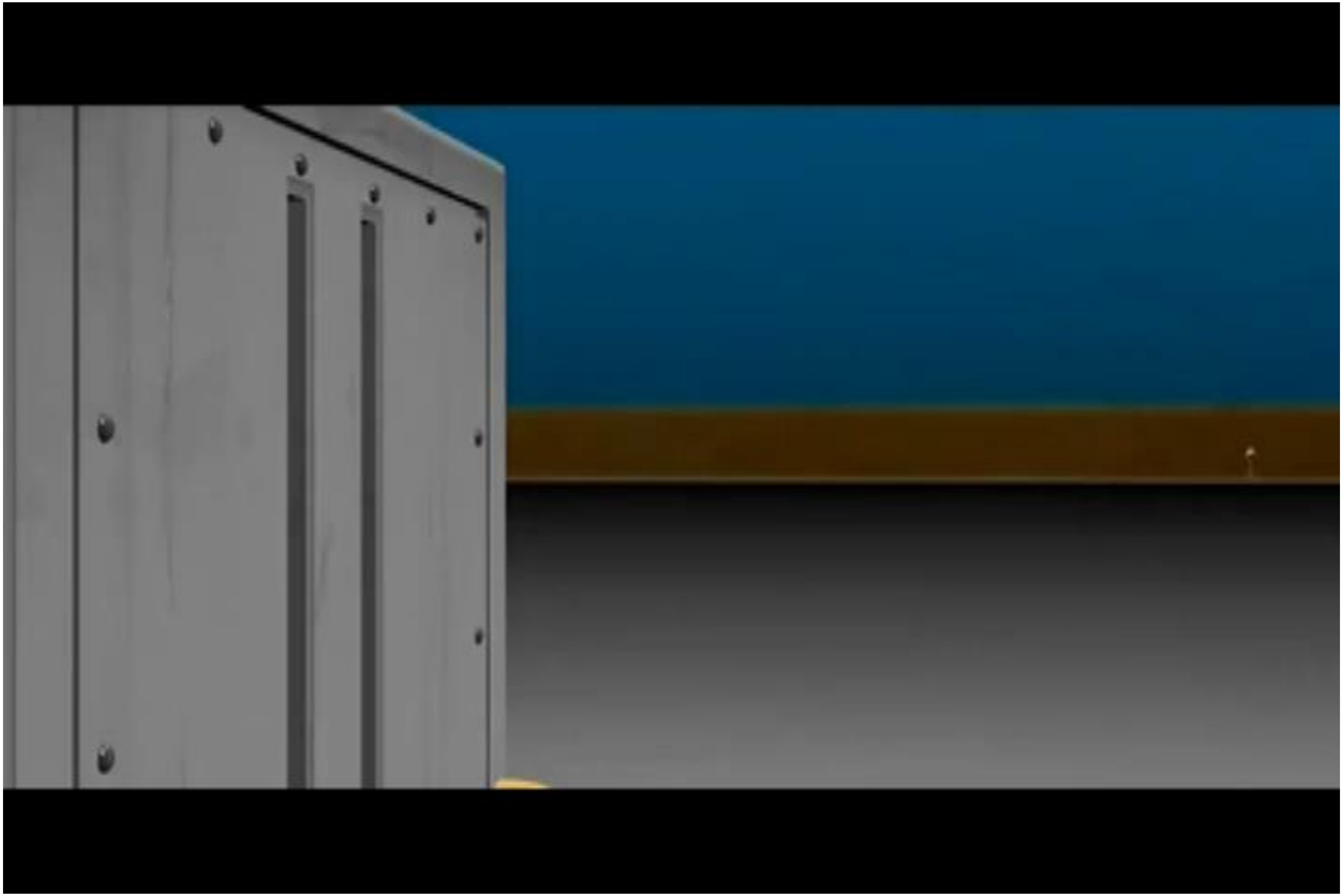
Double Slits – Quantum Electrons



Double Slit – Shoot One Electron At A Time



Double Slits – Quantum Electrons With Observer (Measure At One Slit)



Computational Capacity in the Universe

- Maximum possible elementary quantum logic operations:
  - With gravitational degrees of freedom taken into account

$$\frac{t}{t_p^2} \approx 10^{120}$$

$t \approx 10^{10}$ years is the age of the universe

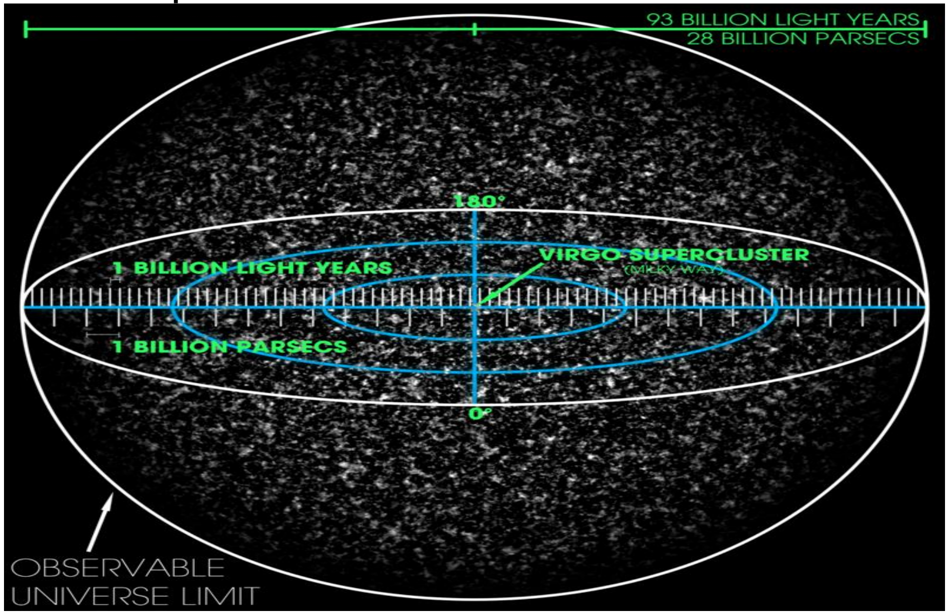
with $t_p = \sqrt{G\hbar / c^5} = 5.391 \times 10^{-44}$ sec

is Planck time (the time scale at which gravitational effects are the same order as the quantum effects)

  - With registered quantum fields alone:

$$\frac{t}{t_p^{3/4}} \approx 10^{90}$$

- Provides *upper bounds* computational capacity performed by all matter since the Universe began
- Provides *lower bounds* of a quantum computer required to simulate the entire Universe required operations and bits
- *If the entire Universe performs a computation*, these numbers give the numbers of operations and bits in that computation



*Seth Lloyd, "Computational Capacity of the Universe", Phys. Rev. Letters, **88**(23), 2002
https://en.wikipedia.org/wiki/Observable_universe

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Quantum Computing Principles

Quantum Principles Important For QC

- Mathematics
  - Primarily Linear Algebra
  - Mathematical Notation – the Dirac Notation

See Backup Slides

- Superposition
- Information Representation
- Uncertainty Principle
- Entanglement
- 6 Postulates of Quantum Mechanics

Quantum Superposition & Uncertainty Principle

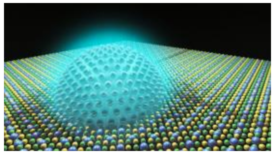
$$\Delta E \Delta t \geq \frac{\hbar}{2}$$

Quantum Information Representation

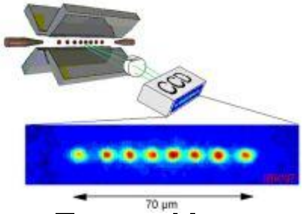
- Physical Representation (Superposition and Entanglement)
  - Electrons Spin Up / Spin Down
  - Nuclear Spins
    - Nuclear Magnetic Resonance
  - Polarization of Light / Photons
  - Optical Lattices
  - Semiconductor Quantum DOT
  - Semiconductor Josephson Junctions
  - Ion Traps
  - Others

- Classical Representation
  - BIT (0,1)

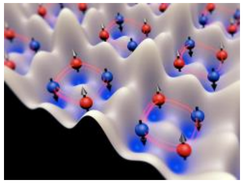
- Quantum Representation
  - Quantum BIT (qubit)



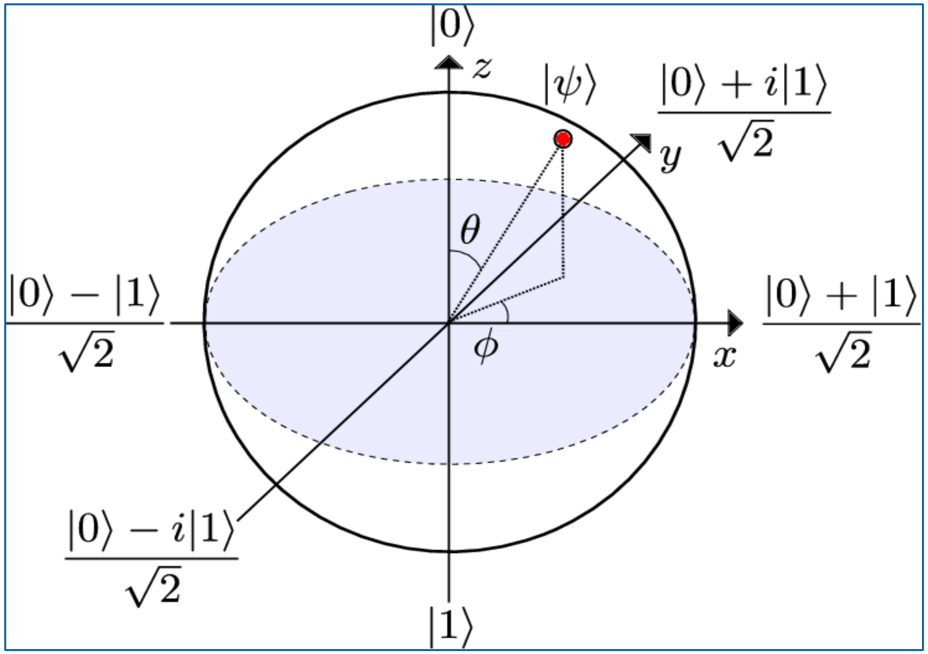
Quantum Dots



Trapped Ions



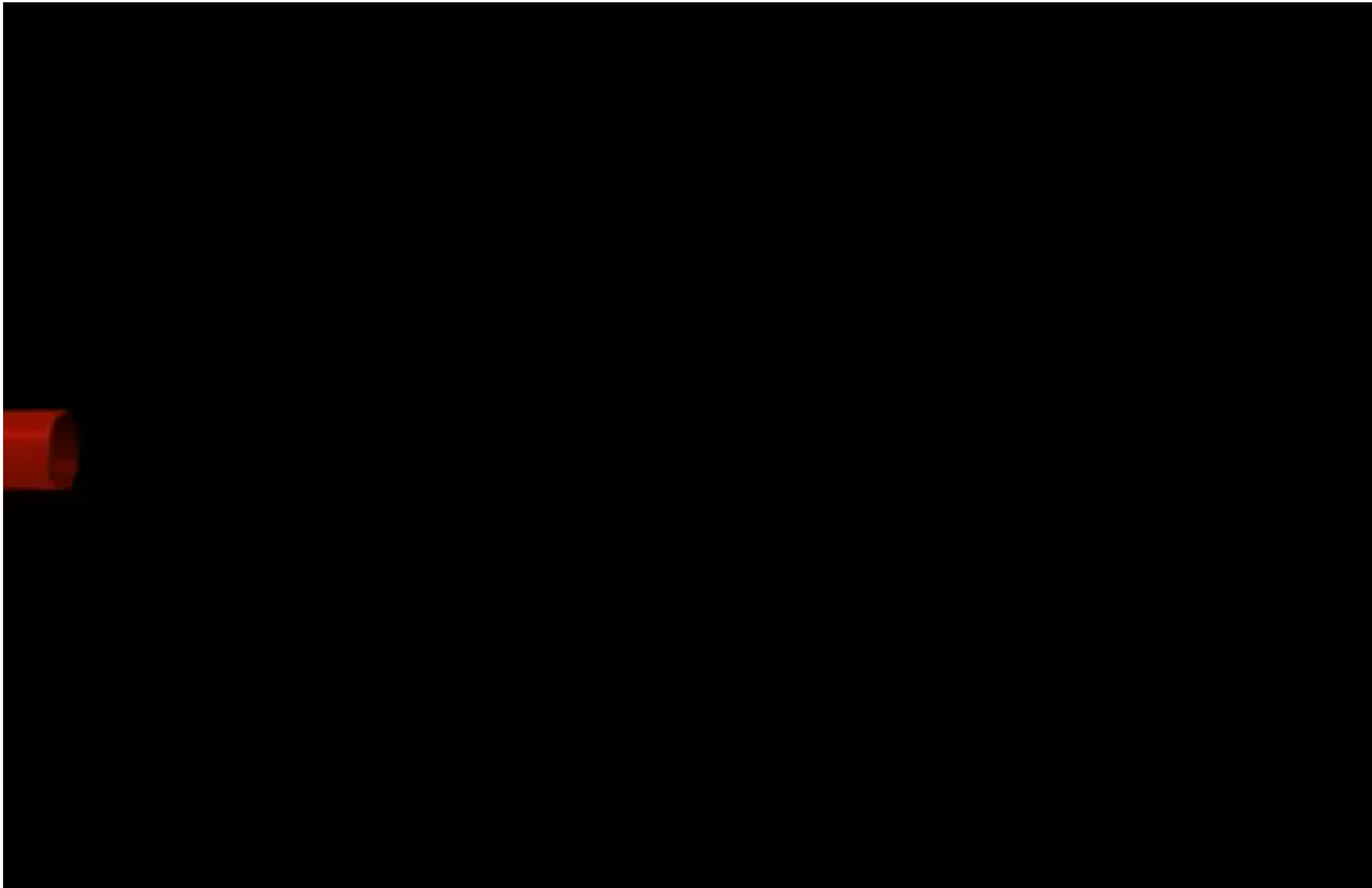
Optical Lattices



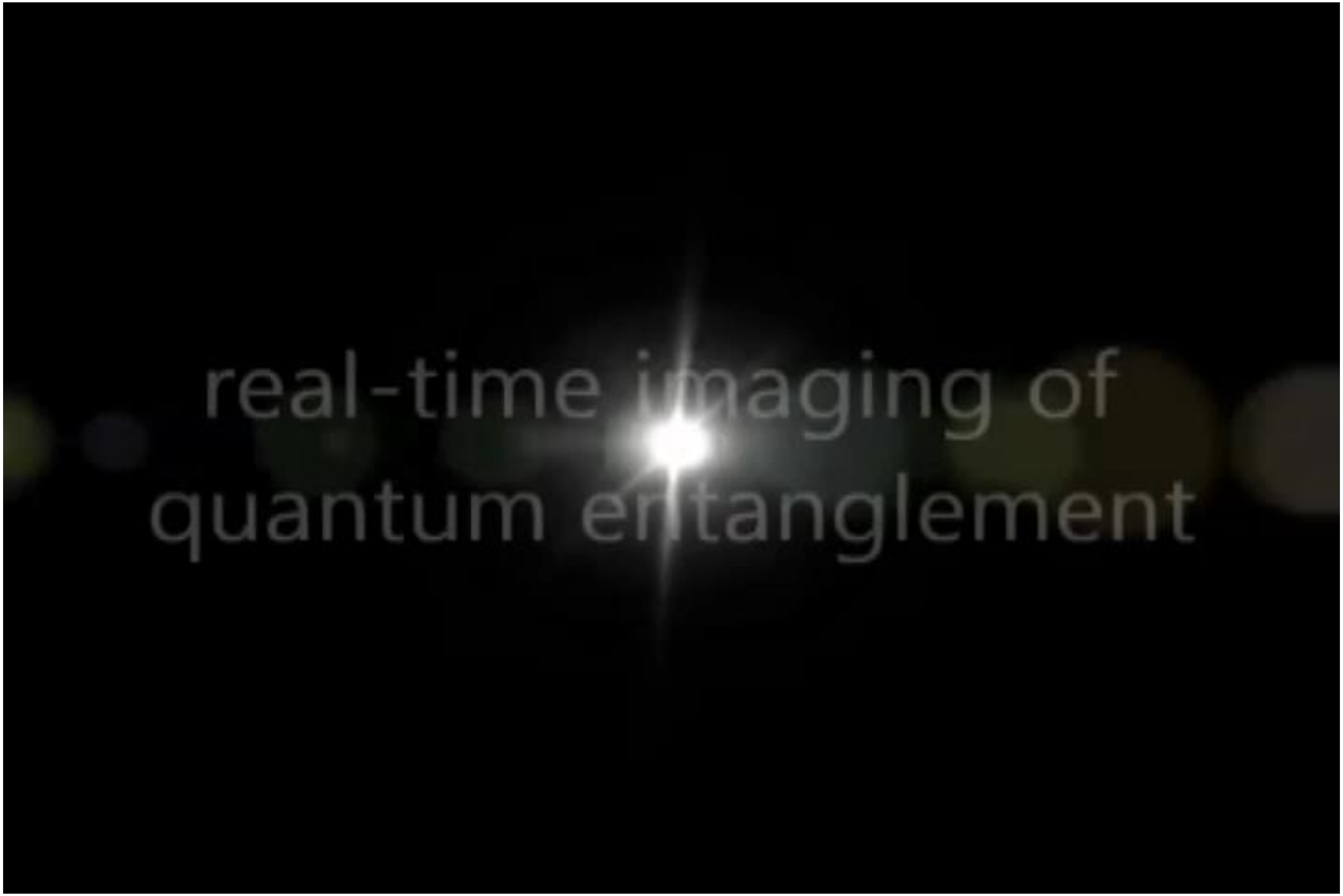
BLOCH representation of a qubit

$$\vec{r}_{BLOCH} = \left(\hat{r}_x, \hat{r}_y, \hat{r}_z\right) = \left(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta\right)$$

Quantum Entanglement
Electrons



Quantum Entanglement
Photons



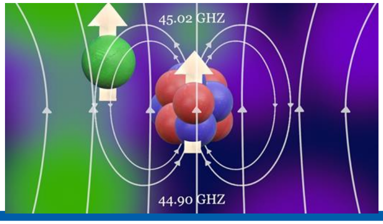
Quantum Entanglement (continued)

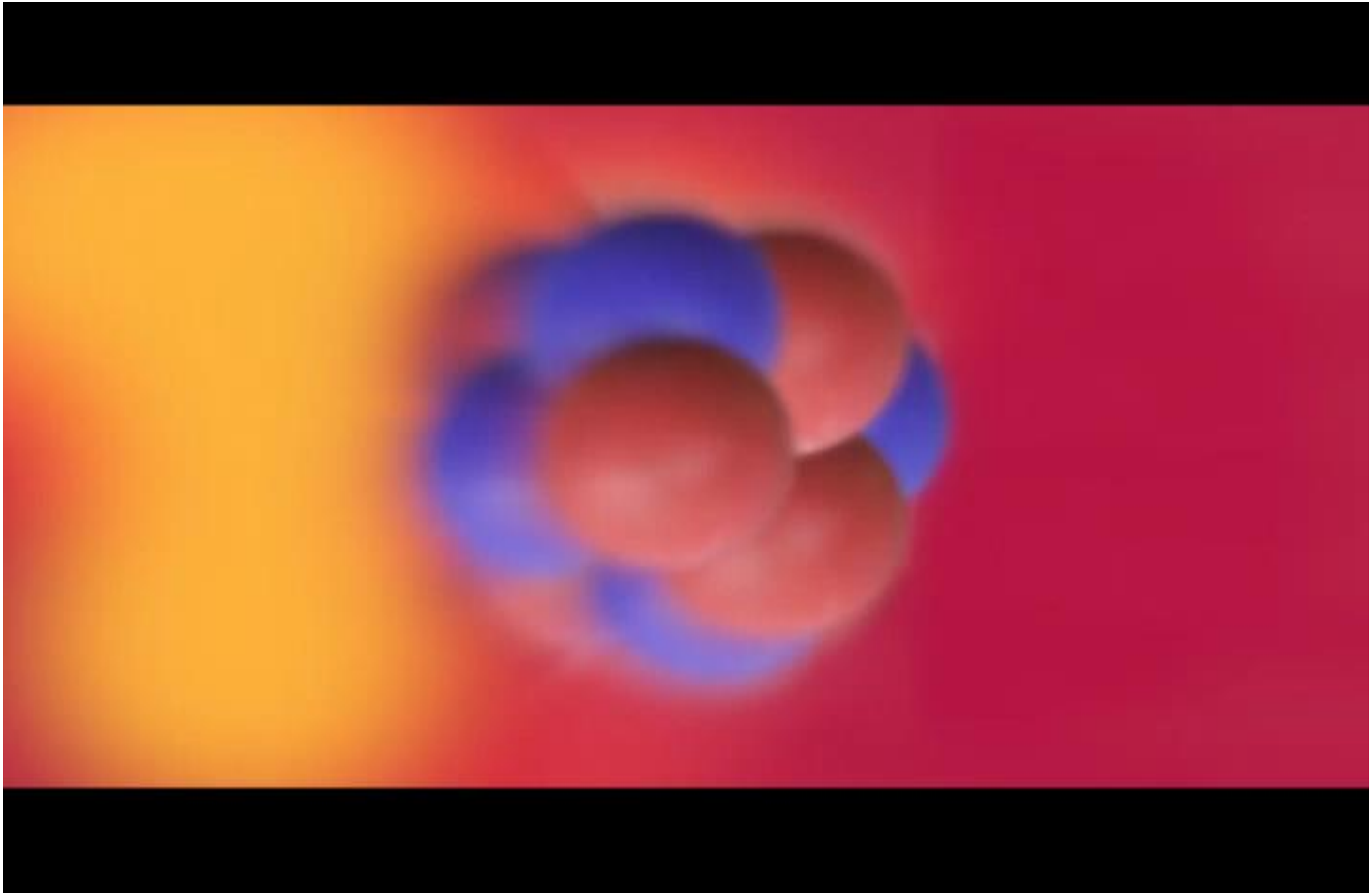
- Start with 2-qubits: $\alpha_0 \,|0\rangle + \alpha_1 \,|1\rangle \; and \; \beta_0 \,|0\rangle + \beta_1 \,|1\rangle$
  - Both are their basis states
- How do we entangle them mathematically?
  - Take the tensor product between the states

$$\left(\alpha_0 \,|0\rangle + \alpha_1 \,|1\rangle\right) \otimes \left(\beta_0 \,|0\rangle + \beta_1 \,|1\rangle\right)$$
$$= \alpha_0\beta_0 \,|00\rangle + \alpha_0\beta_1 \,|01\rangle + \alpha_1\beta_0 \,|10\rangle + \alpha_1\beta_1 \,|11\rangle$$

- 2-qubits in arbitrary states cannot be decomposed into their separate qubit state. As an example, one of the Bell state $|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$, cannot be separated into its individual qubit state
- Einstein called entanglement as “spooky action at a distance,” as it appeared to violate the speed limit of information transmission in theory of relativity (i.e., “c” the velocity of light)

Qubit & Nuclear Spin
Nuclear Magnetic Resonance





6 Postulates of QM deferred to backup slides

Basic Classical & Quantum Computer Operations & Flowchart Of Quantum Control

NORTHROP GRUMMAN

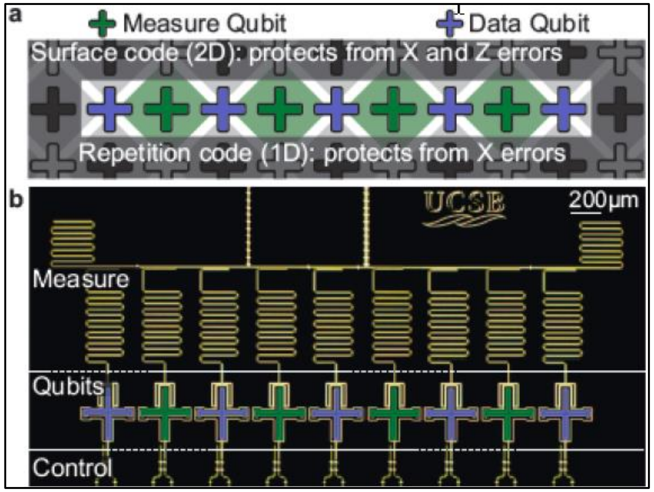
Control of quantum operations

Classical Computer (Master)

Quantum Processor (Slave)

Results of Measurements

Input x n-bit string



Quantum Mechanics Exponential Superposition

Output y n-bit string

$\sum_y \rangle$

May Repeat

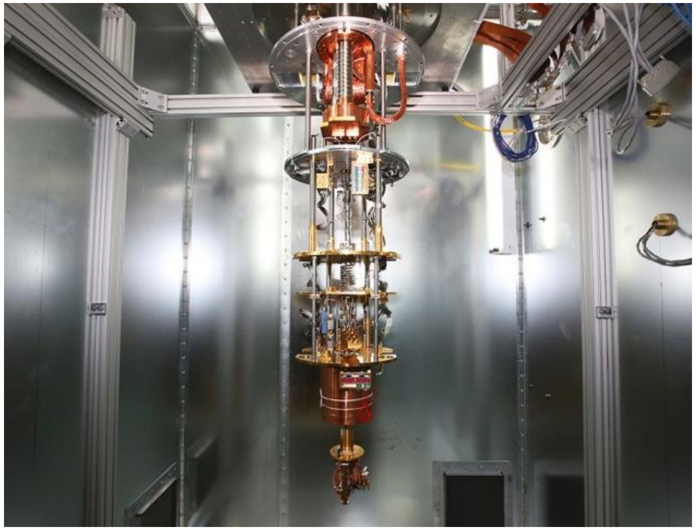
- Classically measurements only reveal n-bits of information
- Probability of 100-bit string y is $|\alpha_y|^2$; new state post measurement is $|y\rangle$

Classical Input → "Quantification"- n qubits in state $|\chi\rangle$ transformed via superposition → Manipulate Apply Gates → Measure $|y\rangle$ → Classical Output

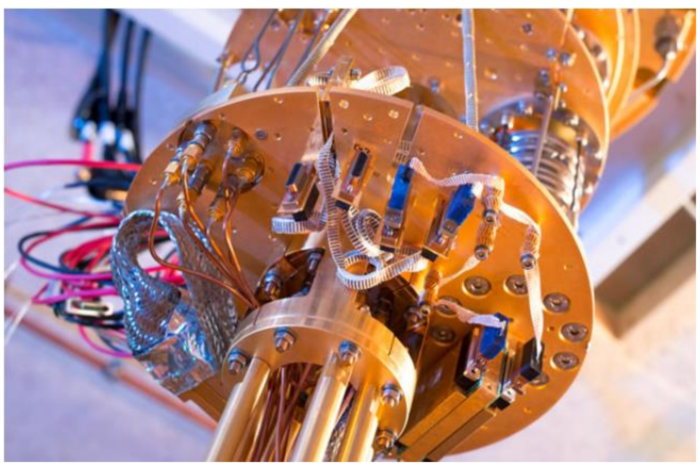
May Repeat

Quantum computer image from: Nature 519, 66–69 (05 March 2015) doi:10.1038/nature14270

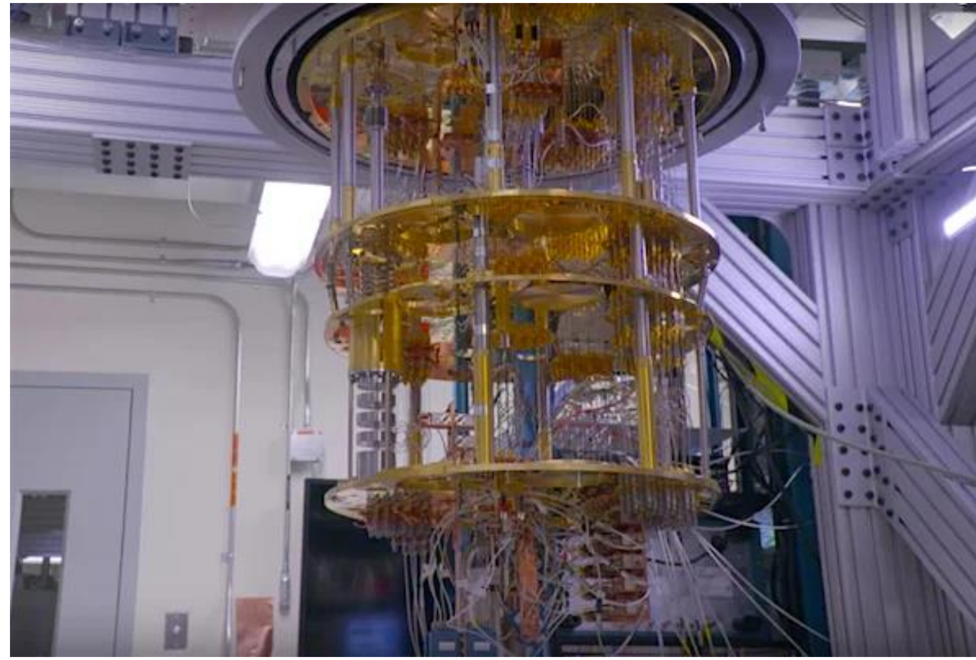
Physical Quantum Computer



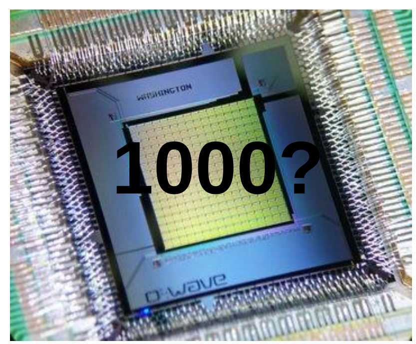
D-Wave



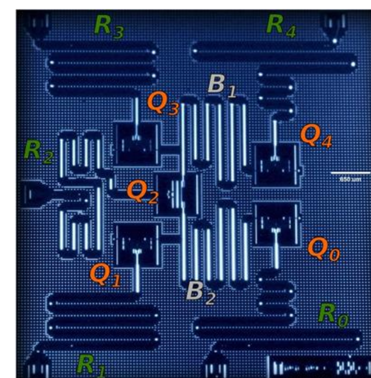
Microsoft



IBM



D-Wave Markets 1000 qubit computers for $10M - $15M

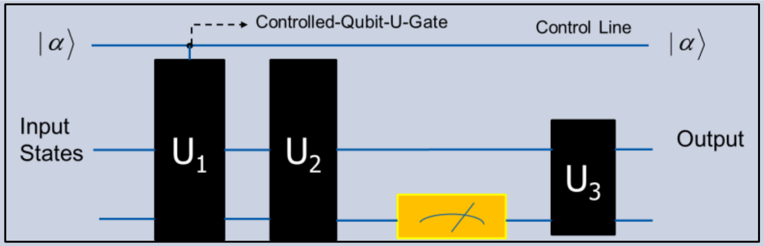
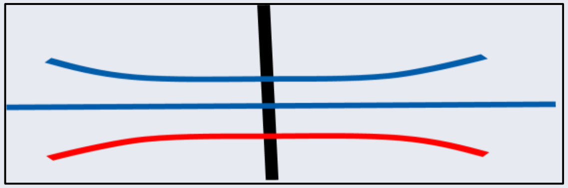


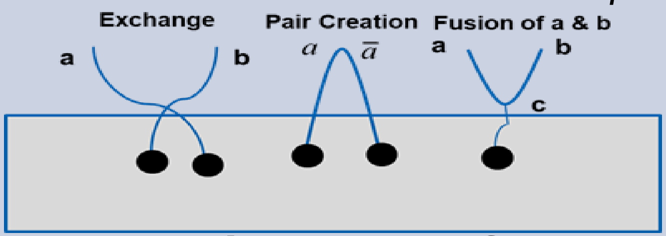
IBM 5 Qubit

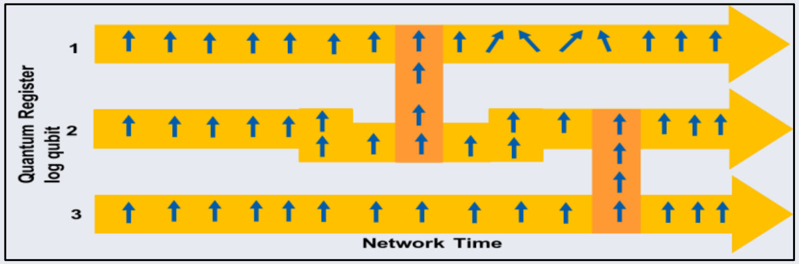
Quantum Computing Models

Models of Quantum Computing

NORTHROP GRUMMAN

<table>
<tr><th>Model</th><th>Description</th></tr>
<tr><td>QC Circuits / Gates</td><td>
</td></tr>
<tr><td>Adiabatic QC
(Vary Hamiltonians slowly from initial to final state)
s = (1/T)</td><td>

$$\hat{H} = (1-s)H_{initial} + sH_{final}$$</td></tr>
<tr><td>Topological QC
(World lines of particles positioned in a plane with time flowing downwards)</td><td>

Computational power of Anyons</td></tr>
<tr><td>Measurement Based QC (Cluster States, Tomography) (local measurement is the only operation needed)</td><td>
</td></tr>
</table>

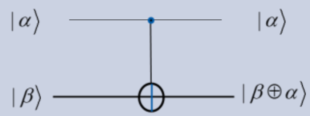
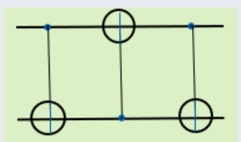
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Quantum Circuits & Gates

Quantum Circuits
Quantum Circuits Error Corrections

NORTHROP GRUMMAN

Gate	Graphical	Mathematical Form	Comments
CNOT		$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$	CNOT gate is a generalized XOR gate: its action on a bipartite state |A,B> is |A, B⊕ A>, where ⊕ is addition modulo 2 (an XOR operation)
SWAP		$\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}$	Swaps states: $(\alpha,\beta)\to(\beta,\alpha)$
Hadamard		$\frac{1}{\sqrt{2}}\begin{pmatrix}1&1\\1&-1\end{pmatrix}=\frac{1}{\sqrt{2}}(\sigma_x+\sigma_z)$	H-gate (square root NOT gate) is an idempotent operator: $H^2 = I$. It transforms the computational basis into equal superpositions.
Pauli X, Y, Z		$\sigma_{X,Y,Z} = \begin{pmatrix}0&1\\1&0\end{pmatrix},\begin{pmatrix}0&-i\\i&0\end{pmatrix},\begin{pmatrix}1&0\\0&-1\end{pmatrix}$	Quantum NOT is identical to σ_x => leaves |0> invariant and changes the sign of |1>. Rotations about the X, Y, Z axis
$\frac{\pi}{8}$ T-Gate		$\begin{pmatrix}1&0\\0&e^{i\frac{\pi}{4}}\end{pmatrix}$	Applies a phase shift to the target qubit. $e^{i\frac{\pi}{4}}\lvert 00\rangle \to$ remains same $e^{i\frac{\pi}{4}}\lvert 11\rangle \to$ target qubit phase shift
Measurement		$\left\{\begin{pmatrix}1&0\\0&0\end{pmatrix},\begin{pmatrix}0&0\\0&1\end{pmatrix}\right\}$	Measurement collapses the superposed quantum states
Qubit		Wire = single qubit	
n Qubits		Wire with n qubits	
Classical bit		Double wire = single bit	

{H, T, and CNOT} are called the "Standard Set." Others in charts below

Properties of Quantum Circuits

- Are not acyclic (no loops)
- No FANIN. This implies that the circuit is not reversible; does not obey unitary operation
- No FANOUT. Cannot copy the qubit's state during the computational phase
  - **No-Cloning Theorem**
    - No copies of qubits in superposition (produces a multipartite entangled state)

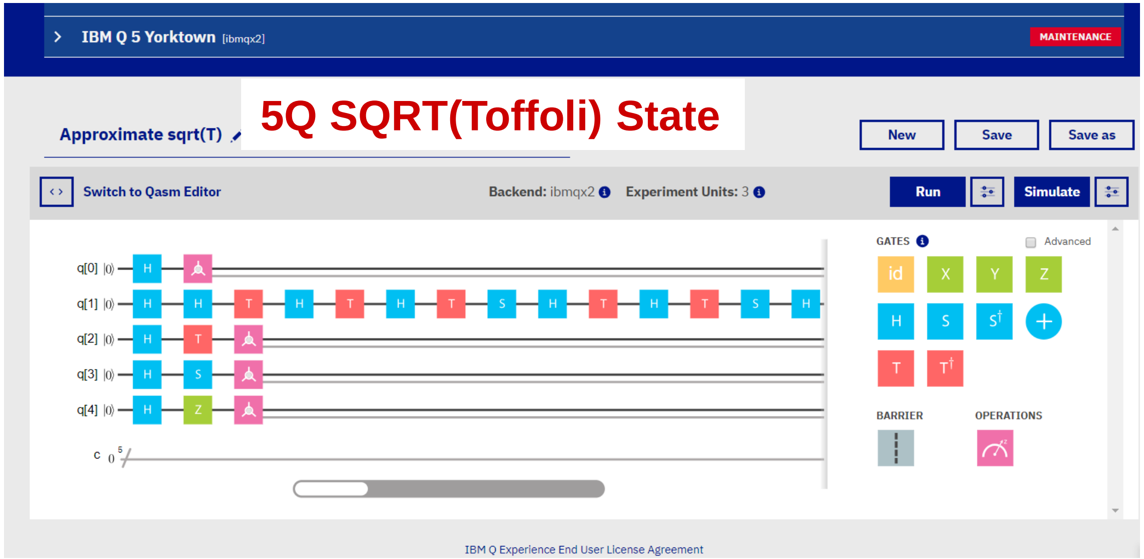
$$|\psi\rangle \underset{NOT\ ALLOWED}{\rightarrow} |\psi\rangle|\psi\rangle|\psi\rangle = \left(\alpha|0\rangle+\beta|1\rangle\right)\otimes\left(\alpha|0\rangle+\beta|1\rangle\right)\otimes\left(\alpha|0\rangle+\beta|1\rangle\right);$$

$$\left(\alpha|0\rangle+\beta|1\rangle\right)\otimes\left(\alpha|0\rangle+\beta|1\rangle\right)\otimes\left(\alpha|0\rangle+\beta|1\rangle\right) = \alpha|000\rangle+\beta|111\rangle = |\psi'\rangle;$$

$$\Rightarrow \text{Entangled 3qubits} \Rightarrow |\psi'\rangle \neq |\psi\rangle$$

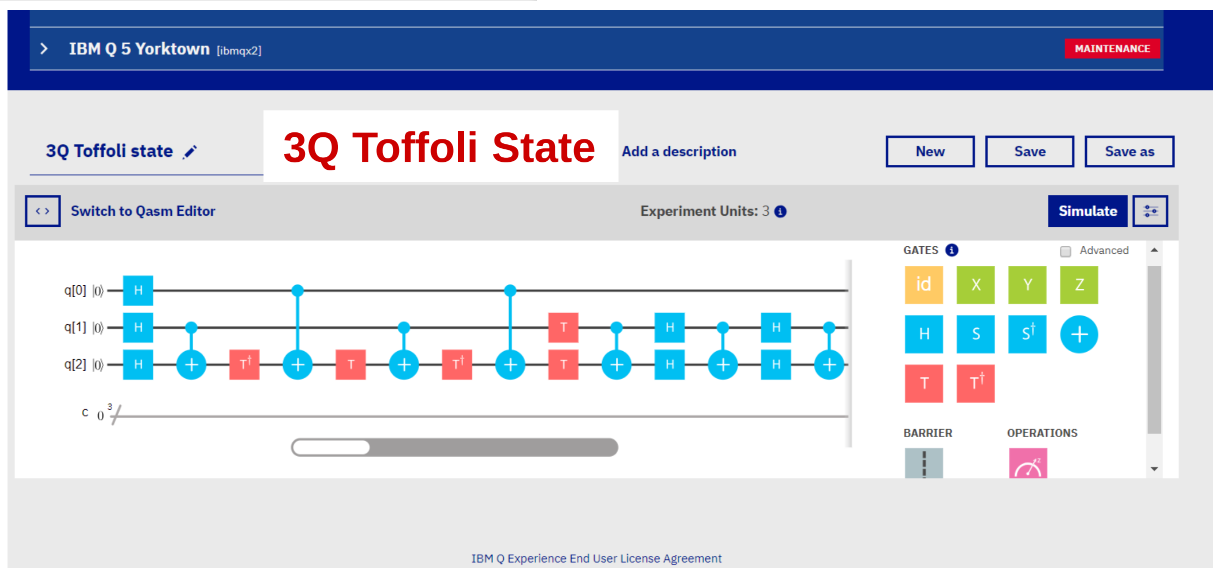
IBM 5-Qubit Quantum Computer Using Toffoli Gates – Freely Available Quantum Computing

NORTHROP GRUMMAN



Timeline for IBM 17-qubit computer is unknown

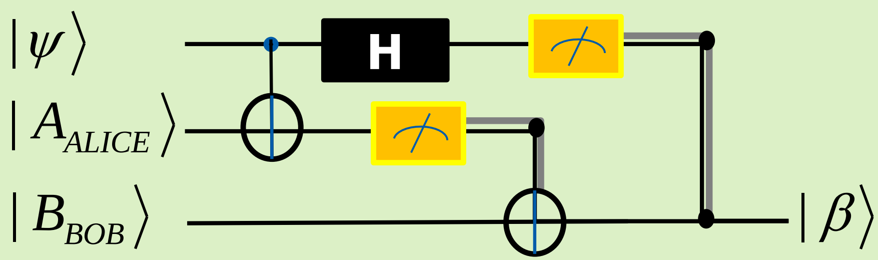
Uses QASM (IBM Q) Assembler or QISKit SDK (Python code) discussed later, for producing the QC circuit results



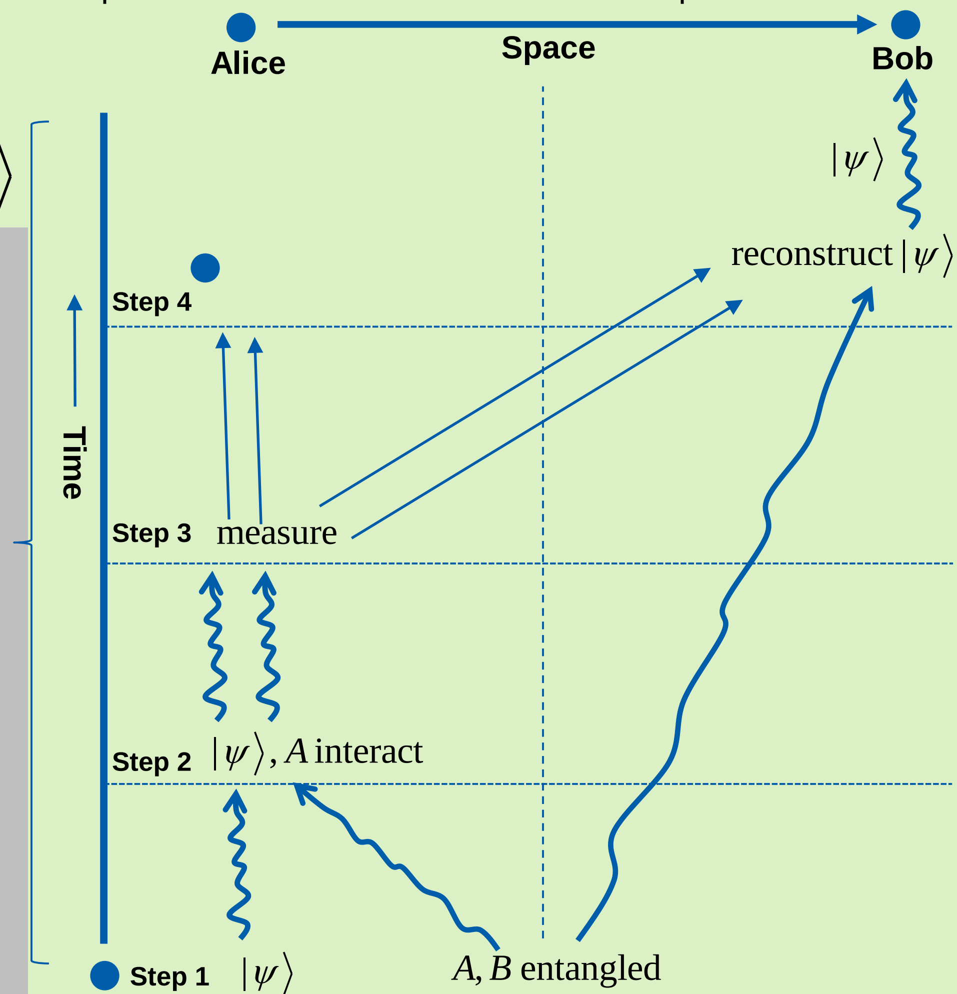
https://quantumexperience.ng.bluemix.net/qx/editor

Multi Qubit Gates (continued)

12) Qubit Teleportation Circuit



An arbitrary qubit is transferred from one location to another. In literature ALICE and BOB example is commonly utilized. Teleportation takes two classical bits to one quantum state.



Squiggly lines correspond to movement of qubits. Straight lines correspond to movement of bits

$|\psi\rangle$ moves from the lower left hand corner from Alice to Bob in the upper right hand corner.

Only two classical bits remain with Alice in Step 4.

SINGLE QUANTUM PARTICLE IS TELEPORTED

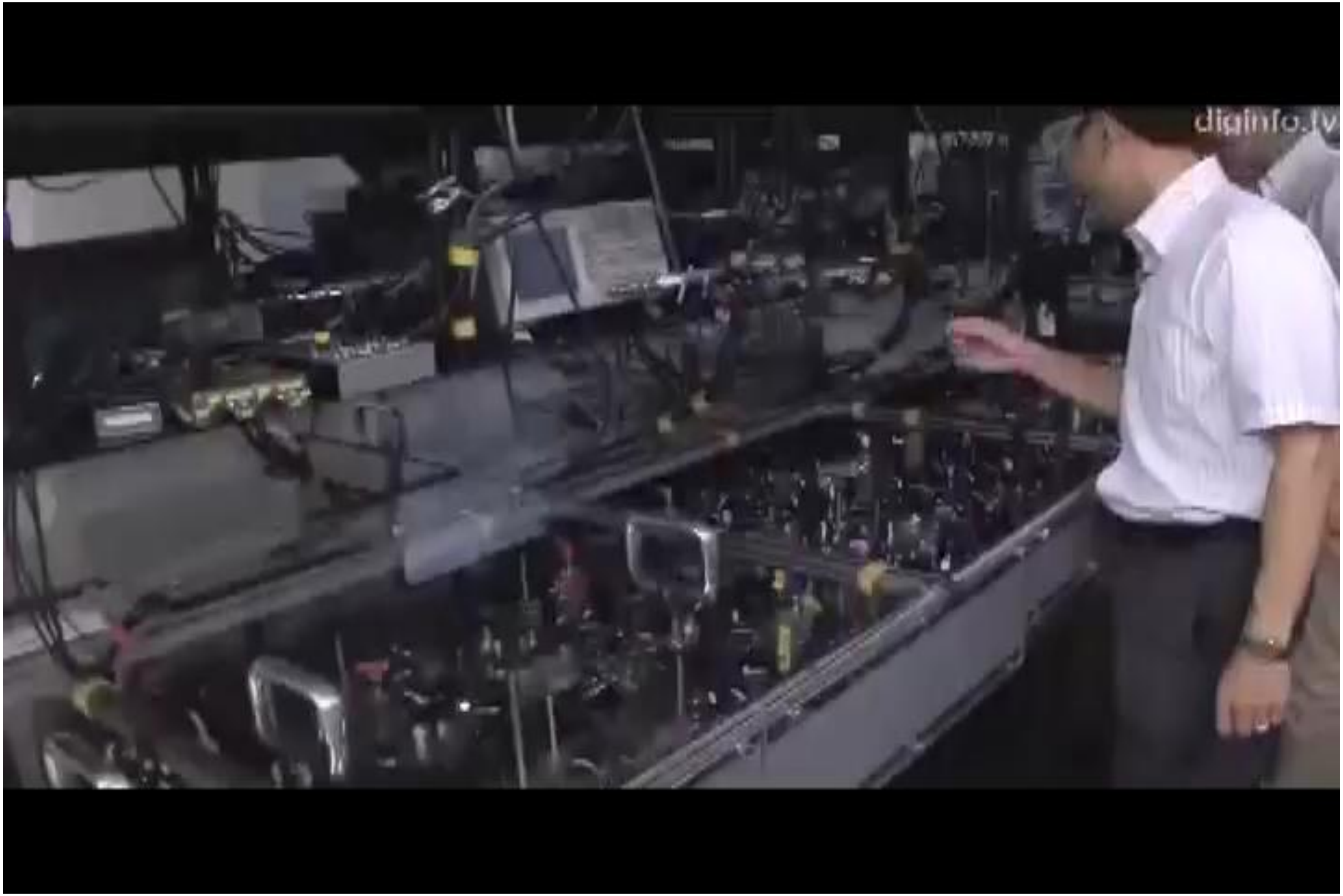
Alice sends (with speed < speed of light) the two classical bits to Bob along a classical channel. Without these Bob will not know what he has received

Entanglement, as well, is not transported faster than the speed of light despite its undisputable magic

Infinite amount of information is passed with the qubit, however once Bob measures he can only get one bit of information

Quantum Teleportation Video

Quantum-Kit Simulation: https://en.wikipedia.org/wiki/Quantum_teleportation#/media/File:Quantum_Teleportation.gif



Quantum Computing Programming Languages

QC Programming Languages and QC Simulators

Product	Description	Website
QCL	C like syntax and complete. The current version of QCL is 0.6.4 (Mar 27 2014), Source Distribution: qcl-0.6.4 *(gcc 4.7 / gnu++98 compliant)*, Binary Distribution (64 bit): qcl-0.6.4-x86_64-linux-gnu.tgz *(AMD64, Linux 3.2, glibc2.13)*	http://tph.tuwien.ac.at/~oemer/qcl.html
QASM	Assembler: Maps directly to quantum circuit model instructions MIT: qasm2circ; QISKit: openQASM	https://www.media.mit.edu/quanta/qasm2circ/ https://qiskit.org/documentation/quickstart.html
QISKit SDK Terra–Python API-Python	QISKit, a quantum program is an array of quantum circuits developed by IBM. Python program code workflow consists of three stages: Build, Compile, and Run.	https://qiskit.org/documentation/quickstart.html https://github.com/QISKit/qiskit-terra https://github.com/QISKit/qiskit-api-py
Q#	Is a C# like quantum programming language developed by Microsoft. It come with a quantum simulator in the quantum development kit.	https://www.microsoft.com/en-us/quantum/development-kit
CodeProject	Is a Java quantum code project	https://www.codeproject.com/Articles/1130092/Java-based-Quantum-Computing-library
Quantum-Kit	Is a graphical quantum circuit simulator	https://sites.google.com/view/quantum-kit/home
Other Simulators in various languages and tools	C/C++, CaML, Ocaml, F#, GUI based, Java, Javascript, Julia, Maple, Mathematica, Maxima, Matlab/Octave, .NET, Online Services, Perl/PH,P, Python, Rust, and Scheme/Haskell/LISP/ML	https://quantiki.org/wiki/list-qc-simulators
Other Languages	See Wikipedia	https://en.wikipedia.org/wiki/Quantum_programming

Simple QSAM Example Using QRAM Model

NORTHROP GRUMMAN

INITIALIZE R 2
U TENSOR H H
APPLY U R
MEASURE R RES

Allocates register R to 2 qubits and initializes to $|00\rangle$

$$U = H \otimes H = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

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

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

Measures the q-register R and stores in bit array RES. What is the probability for the ground state (i.e., expectation value)?

$$\text{from coefficient of } |00\rangle : \left|\frac{1}{2}\right|^2 = \frac{1}{4} = 0.25$$

Parallel application of H to the 2-qubits puts the register R in a balanced superposition of four basis states

Quantum Algorithms

Quantum Computing Algorithms (continued)

Algorithm	Description	Reference
Algorithms Based on QFT		
Shor's; $O\left(n^2(\log N)^3\right)$	Integer factorization (given integer N find its prime numbers); discrete logarithms, hidden subgroup problem, and order finding	Peter W. Shor, "Algorithms for Quantum Computation Discrete Log and Factoring," AT&T Bell Labs, shor@research.att.com
Simon's; *exponential*	Exponential quantum-classical separation. Searches for patterns in functions	Simon, D.R. (1995), *"On the power of quantum computation"*, Foundations of Computer Science, 1996 Proceedings., 35th Annual Symposium on: 116–123, retrieved 2011-06-06
Deutsch's, Deutsch's – Jozsa, an extension Deutsch's algorithm	Depicts quantum parallelism and superposition. "Black Box" inside. Can evaluate the input function, but cannot see if the function is balanced or constant	David Deutsch (1985). "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". Proceedings of the Royal Society of London A. 400: 97 David Deutsch and Richard Jozsa (1992). "Rapid solutions of problems by quantum computation". Proceedings of the Royal Society of London A. 439: 553
Bernstein/Vazirani; *polynomial*	Superpolynomial quantum-classical separation	Ethan Bernstein and Umesh Vazirani. *Quantum complexity theory*. In Proc. 25th STOC, pages 11–20, 1993
Kitaev	Abelian hidden subgroup problem	A. Yu. Kitaev. *Quantum measurements and the Abelian stabilizer problem*, arXiv:quant-ph/9511026, 1995
van Dam/Hallgren	Quadratic character problems	Wim van Dam, Sean Hallgren, *Efficient Quantum Algorithms for Shifted Quadratic Character Problems.* CoRR quant-ph/0011067 (2000)
Watrous	Algorithms for solvable groups	John Watrous, Quantum algorithms for solvable groups, **arXiv:quant-ph/0011023**, (2001)
Hallgren	Pell's equation	Sean Hallgren. *Polynomial-time quantum algorithms for pell's equation and the principal ideal problem,* Proceedings of the thirty-fourth annual ACM symposium on the theory of computing, pages 653–658. ACM Press, 2002.
Algorithms Based on Amplitude Amplification		
Grover's; $O\left(\sqrt{N}\right)$	Search algorithm from an unordered list (database) for a marked element, and statistical analysis	Lov Grover, *A fast quantum mechanical algorithm for database search,* In Proceedings of 28th ACM Symposium on Theory of Computing, pages 212–219, 1996
Traveling Salesman Problem; $O\left(\sqrt{N}\right)$	Special case of Grover's algorithm	https://en.wikipedia.org/wiki/Travelling_salesman_problem

Machine Learning **Quantum Particle Swarm Optimization (QPSO)**

Quantum Algorithms (continued)
Machine Learning Applications

Quantum Algorithm	Grover's Algorithm Applied?*	Execution Improvement vs. Classical	Quality of Learning Algorithm Studied	Quantum Computer Implementation	Quantum States?#	Reference
Neural networks	Yes		Numerical	Yes	No	1
Boosting	No	Quadratic	Analytical	Yes	No	2
K-medians	Yes	Quadratic	No	No	No	3
K-means	Optional	Exponential	No	No	Yes	4
Principal components	No	Exponential	No	No	Yes	5
Hierarchical clustering	Yes	Quadratic	No	No	No	6
Associative memory	Yes No		No No	No No	No	7 8
Support vector machines	Yes No	Quadratic Exponential	Analytical No	No No	No Yes	9 10
Nearest neighbors	Yes	Quadratic	Numerical	No	No	11
Regression	No		No	No	Yes	12
Hidden Markov Chains	No		No	No	No	13
Bayesian Methods	No		No	No	No	14

*Grover's search or extension used; #Input or output were both quantum states vs. classical vectors

Most topics from: Peter Wittek, "Quantum Machine Learning", Elsevier Insights, 2014

Summary

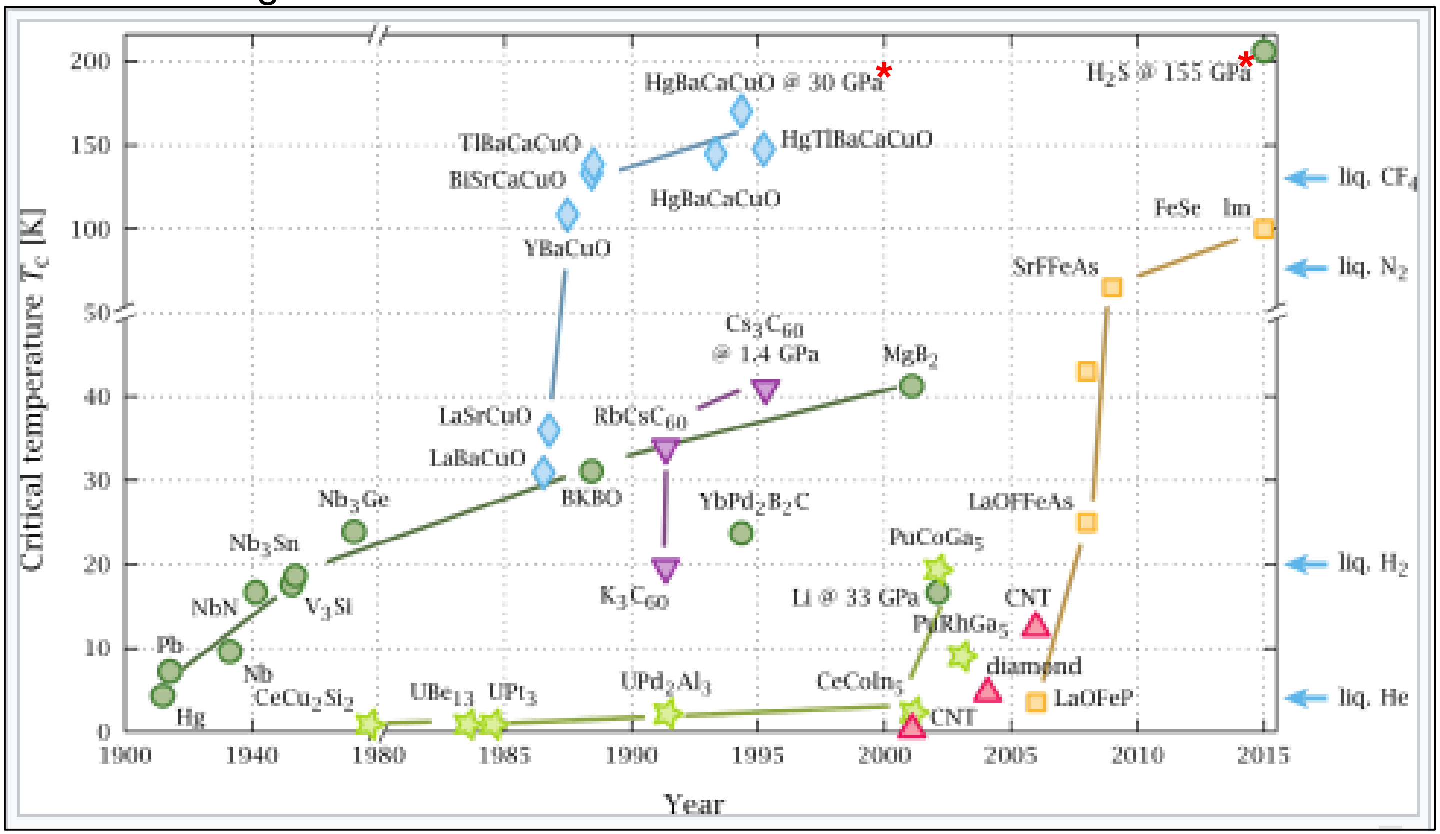
Summary

- Develop / reuse quantum gates tailored to PHM
  - Creating “Oracles” are a very useful technique
  - Note: QC gates in series accumulate errors (described earlier)
- Tailor quantum machine learning algorithms for PHM algorithms
- “Quantum particle swarm optimization (QPSO)” appears to be a good candidate for dynamic degraded state prognostics (tracks dynamic changes to a particle in its local focus specified by the characteristics length vector of the swarm in some Hamiltonian potential $\{E+V(\vec{r})\}$. Implementation on a quantum computer? Develop technology / gates / methods to do QPSO on quantum computers.
- Consider “adiabatic quantum computing” as an alternative approach. It is based on the time evolution of a quantum system. A quantum adiabatic process is one in which the initial Hamiltonian evolves slowly to it final Hamiltonian (i.e., the time scale should be proportional to the energy difference between the ground state and the first excited)
  - For electrons the Hamiltonian can be represented by the Pauli operators
- Consider “cluster state quantum computing” that does not rely on quantum gates to do its processing (multi-qubits operations)

Treatment of topics discussed here are in Backup Slides

Summary (continued)

- Develop quantum computer with superconducting materials at higher temperatures
  - Apart for space vehicles, current implementations for temperatures < 1.5°K are not feasible out of large infrastructures



*1 Gigapascal = 9869.2 Atmosphere

https://en.wikipedia.org/wiki/Superconductivity & Ref. 15, 16

Summary (continued)

NORTHROP GRUMMAN

- Is classical cybersecurity safe with the power of quantum computing?
  - Reference: Lily Chen et al., “Report on Post-Quantum Cryptography”, 2016 http://dx.doi.org/10.6028/NIST.IR.8105

Table 1 - Impact of Quantum Computing on Common Cryptographic Algorithms

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	---------------	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

  - If DWave 1000/2000 qubits Quantum Computer is a reality, AES and SHA-2/SHA-3 are unsafe

- Is quantum computing cybersecurity safe?
  - It is possible that we would need methods / techniques to keep Quantum Computers safe?
    - Still the issue of classical measurements

Treatment of topics discussed here are in Backup Slides

References

- Videos
  - [Video from https://www.youtube.com/watch?v=fwXQjRBLwsQ](https://www.youtube.com/watch?v=fwXQjRBLwsQ) (Slits Video)
  - https://www.youtube.com/watch?v=815oMDT5g0o (Superposition Video)
  - https://www.youtube.com/watch?v=9lOWZ0Wv218 (Entanglement Video)
  - https://www.youtube.com/watch?v=zNzzGgr2mhk (Nuclear Magnetic Resonance Video)
  - https://www.youtube.com/watch?v=f5vOfr1dl4o (Teleportation Video)
- Quantum Mechanics Books
  - *Dirac Notation: “Principles of Quantum Mechanics”, Ramamurti Shankar, Plenum Press, New York / London, 1980*
  - *“Lectures on Quantum Mechanics”, Steven Weinberg, Cambridge University Press, New York, 2013*
  - *“Lectures on Computation”, Richard P. Feynman, Westview Press (1996, reprinted 1999)*
- *Quantum Computing Books*
  - *Jun Sun, Choi-Hong Lai, Xiao-Jun Wu, “Particle Swarm Optimisation-Classical and Quantum Perspective”, Chapman & Hall/CRC Press,2012*
  - *Peter Wittek, “Quantum Machine Learning”, Elsevier Insights, 2014*

Papers

1. Narayanan and Menneer (2000), *Quantum artificial neural network architectures and components*, Inform Sci., 128(3-4), 231-255
2. Neven et al. (2009), *Quantum pattern recognition with liquid-state nuclear magnetic resolution*, Phys. Rev. A **79**, 042321
3. Aimeur et al. (2013), *Quantum speed-up for unsupervised learning Mach. Learn*., 90(2), 261-287
4. Lloyd et al. (2013), *Quantum algorithms for supervised and unsupervised machine learning*, arXiv:1307.0411
5. Lloyd et al. (2013), *Quantum principle component analysis*, arXiv:1307.0411
6. Aimeur et al. (2013), *Quantum speed-up for unsupervised learning Mach. Learn*., 90(2), 261-287
7. Ventura and Martinez (2000), *Quantum associative memory*, Infom. Sci., 124(1), 273-296
8. Trugenberger (2001), *Probabilistic quantum memories*, Phys. Lett., 87, 067901
9. Anguita et al. (2003), *Quantum optimization for training support vector machines,* Neural Netw, 16(5), 763-770
10. Rebentrost et al. (2013), *Quantum support vector machine for big feature and big feature classification*, arXiv:1307.0471
11. Wiebe et al. (2014), *Quantum nearest neighbor algorithms for machine learning*, arXiv:1401.2142
12. Bisio et al. (2010), *Optimal quantum learning of a unitary transformation*, Phys. Rev. A 81(3), 032324

References (continued)

13 Siddarth Srinivasan et al., *Learning Hidden Quantum Markov Models*, Proceedings of the 21stInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. JMLR: W&CP volume 7X

14 Sentís, Gael; Calsamiglia, John; Muñoz-Tapia, Raúl; Bagan, Emilio (2012), *Quantum learning of coherent states*. *EPJ Quantum Technology*. **2** (1). doi:10.1140/epjqt/s40507-015-0030-4

15 Zhi-An Ren; et al. (2008), *Superconductivity at 55 K in iron-based F-doped layered quaternary compound Sm[$O_{1-x}F_x$]FeAs*, "Chin. Phys. Lett. 25, 2215 (2008)

16. Li, Yinwei; Hao, Jian; Liu, Hanyu; Li, Yanling; Ma, Yanming (2014-05-07), "The metallization and superconductivity of dense hydrogen sulfide". *The Journal of Chemical Physics*. **140** (17): 174712. arXiv:1402.2721

17 Zhi-An Ren; et al. (2008),), *Superconductivity at 55 K in iron-based F-doped layered quaternary compound Sm[$O_{1-x}F_x$]FeAs*, "Chin. Phys. Lett. 25, 2215 (2008)

18 Gelo Noel M. Tabia (2011), *Qutrits Under a Microscope*, American Physical Society March Meeting 2011, Dallas, Texas, United States (21 - 25 March 2011)

19 Frank Wilczek (1982), Phys. Rev. Lett. **49**, 957

20 A. Kitaev (2003), Ann. Phys. **303,** 2

21 Jacobson, Nathan (2009). *Basic Algebra I* (2nd ed.). Dover Publications

- *Various Quantum Computing Websites*
  - *https://en.wikipedia.org/wiki/Quantum_computing*
  - *https://qnncloud.com/*
  - *https://en.wikipedia.org/wiki/Quantum_machine_learning*
  - *https://en.wikipedia.org/wiki/Quantum_algorithm*

Backup Slides

Quantum Mathematics

NORTHROP GRUMMAN

- Mathematics
  - Primarily Linear Algebra
  - Notation Dirac Notation

$$"Bra"\langle\psi|; \quad "Ket" \;|\psi\rangle;$$

$$\langle\psi| \;=\; |\psi\rangle^{\dagger} = |\psi^{*}\rangle^{T} \;; \dagger \text{ is Ajoint Operator}$$

$$\langle\psi|\psi\rangle = 1 = \int dx\psi^{*}(x)\psi(x);$$

$$\left(|\psi\rangle, |\phi\rangle\right) = \langle\psi|\phi\rangle = \int dx\, \psi^{*}(x)\phi(x);$$

$$\langle\psi|\phi\rangle^{*} = \langle\phi|\psi\rangle;$$

$$\langle\psi|\hat{H}|\psi\rangle = \int dx\, \psi^{*}(x)\hat{H}\psi(x)$$

$$(Acting\ on\ an\ Hamiltonian);$$

$$Schrödinger\ Hamiltonian\ for\ the$$

$$N - particle\ case\ (\hbar = 6.626\times10^{-34}\ Joule\,\text{sec}):$$

$$\hat{H} = \frac{-\hbar}{2}\sum_{n=1}^{N}\frac{1}{m_n}\nabla_n^2 + V(\vec{r}_1, \vec{r}_2 \ldots \vec{r}_N, t);$$

$$Time\ dependent\ Schrödinger\ Equation$$

$$i\,\hbar\frac{\partial}{\partial t}\Psi(\vec{r},t) = \hat{H}\Psi(\vec{r},t)$$

More mathematical details in Backup Slides

Quantum Mathematics

- Mathematics
  - Primarily Linear Algebra
  - Notation Dirac Notation

$$"Bra"\langle\psi|;\quad "Ket"\ |\psi\rangle;$$

$$\langle\psi| = |\psi\rangle^{\dagger} = |\psi^{*}\rangle^{T};\dagger \text{ is Ajoint Operator}$$

$$\langle\psi|\psi\rangle = 1 = \int dx\psi^{*}(x)\psi(x);$$

$$\left(|\psi\rangle, |\phi\rangle\right) = \langle\psi|\phi\rangle = \int dx\,\psi^{*}(x)\phi(x);$$

$$\langle\psi|\phi\rangle^{*} = \langle\phi|\psi\rangle;$$

$$\langle\psi|\hat{H}|\psi\rangle = \int dx\,\psi^{*}(x)\hat{H}\psi(x)$$

$(Acting\ on\ an\ Hamiltonian);$

$Schrödinger\ Hamiltonian\ for\ the$

$N-particle\ case:$

$$\hat{H} = \frac{-\hbar}{2}\sum_{n=1}^{N}\frac{1}{m_n}\nabla_n^2 + V(\vec{r}_1,\vec{r}_2\ldots\vec{r}_N,t)$$

$Time\ dependent\ Schrödinger\ Equation:$

$$i\,\hbar\frac{\partial}{\partial t}\Psi(\vec{r},t) = \hat{H}\Psi(\vec{r},t)$$

Linear Combination: $|a\rangle = \sum_{i=1}^{n} c_i\,|b_i\rangle;$

Linear Independence: $\sum_{i=1}^{n} c_i\,|b_i\rangle = 0\ iff\ c_1 = \ldots = c_n = 0;$

Probability Amplitudes: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \Rightarrow$

$$|\alpha|^2 + |\beta|^2 = 1;$$

Norm: $\||\alpha\rangle\| = \sqrt{\langle\alpha|\alpha\rangle} = \sqrt{\sum_{i=1}^{n}|\alpha_i|^2}$; unit vector: $\sum_i|\alpha_i|^2 = 1;$

Inner Product: $\langle\alpha|\alpha'\rangle = \left(\alpha_1^{*},\ \ldots\ ,\alpha_n^{*}\right)\begin{pmatrix}\alpha_1' \\ \ldots \\ \alpha_n'\end{pmatrix}; \langle\alpha|\beta\rangle = \langle\beta|\alpha\rangle^{*}; \langle\alpha|\beta\rangle = \sum_{i}^{n}\alpha_i^{*}\beta_i;$

Outer Product: $|\alpha\rangle\langle\beta| = \begin{pmatrix}\alpha_1 \\ \ldots \\ \alpha_n\end{pmatrix}\left(\beta_1^{*},\ \ldots\ ,\beta_n^{*}\right);$

Tensor Product: $|\alpha\rangle\otimes|\beta\rangle = |\alpha\rangle|\beta\rangle = |\alpha\beta\rangle;$

$$A\otimes B = \begin{pmatrix}a & b \\ c & d\end{pmatrix}\otimes\begin{pmatrix}x & y \\ v & w\end{pmatrix} = \begin{pmatrix}ax & ay & bx & by \\ av & aw & bv & bw \\ cx & cy & dx & dy \\ cv & cw & dv & dw\end{pmatrix};$$

Orthogonality: $\langle\alpha|\beta\rangle = 0;$

Orthonormality: $\langle\alpha|\beta\rangle = \delta_{ij}\quad (i,j = 1,2,\ldots,n); \delta_{ij} = 0, i \neq j;$

Trace: $tr(\alpha) = \sum_{i=1}^{n}\alpha_{ii};$

Hermitian Operators: $\psi^{\dagger} = \psi;$

$$\psi^{\dagger} = -\psi\ (anti)$$

6 Postulates of Quantum Mechanics

NORTHROP GRUMMAN

- Postulate 1: *At each instant the state of a physical system is represented by a ket* $|\psi\rangle$ *in the space of states*

- Postulate 2: *Every observable attribute of a physical system is described by an operator that acts on the kets that describe the system*

$$\hat{A}:|\psi\rangle \to |\psi'\rangle = \hat{A}|\psi\rangle$$

  - *For every operator, there are special states that are not changed (except for being multiplied by a constant) by the action of an operator*

$$\hat{A}:|\varphi_a\rangle = a|\varphi_a\rangle$$

$$\varphi_a \text{ are eigenstates}$$

$$a \text{ is eigenvalue}$$

6 Postulates of Quantum Mechanics (continued)

- Postulate 3: *The only possible result of the measurement of an observable* A *is one of the eigenvalues of the corresponding operator* $\hat{A}$

- *Postulate 4: When a measurement of an observable A is made on a generic state* $|\psi\rangle$*, the probability of obtaining an eigenvalue* a_n *is given by the square of the inner product of* $|\psi\rangle$ *with the eigenstate* $|a_n\rangle$ *is* $\left|\langle a_n|\psi\rangle\right|^2$, $\langle a_n|\psi\rangle$ *is the probability amplitude*

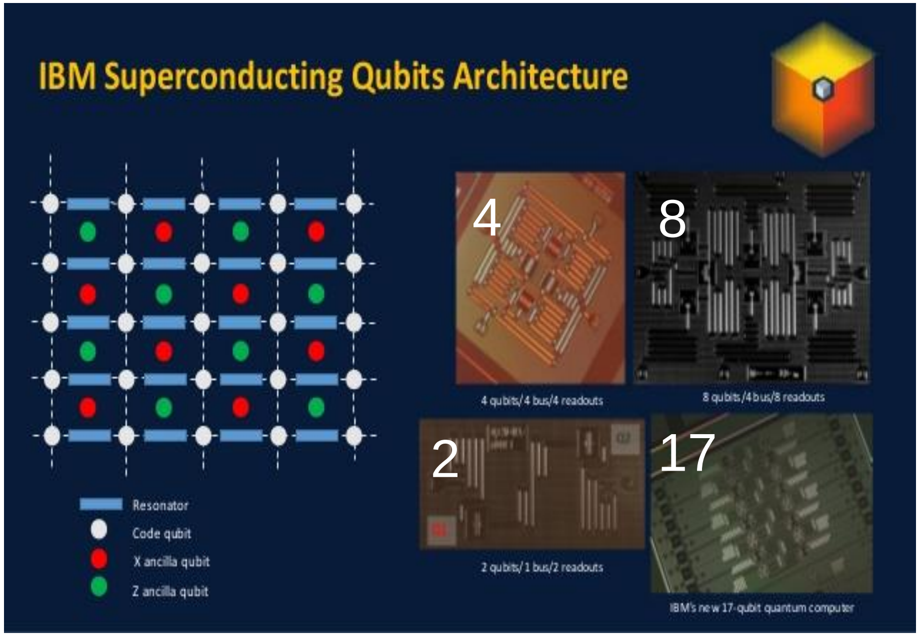
6 Postulates of Quantum Mechanics (continued)

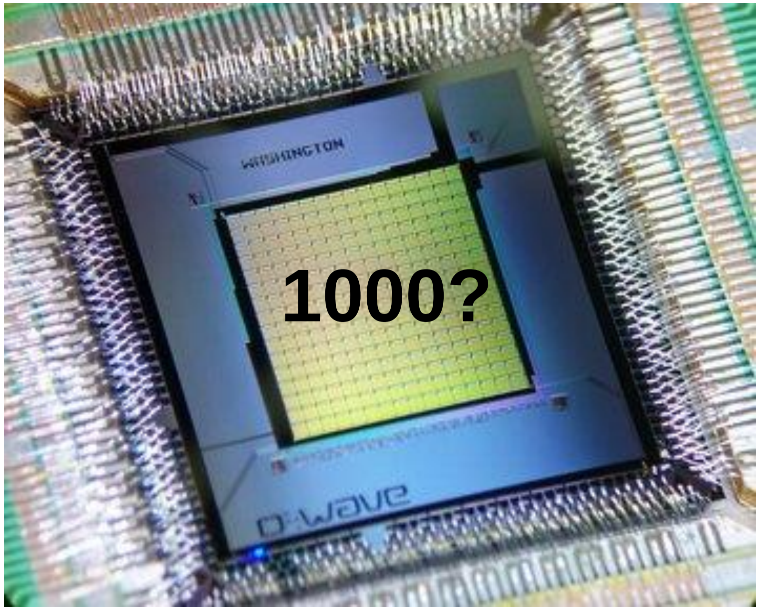
- Postulate 5: *Immediately after the measurement of an observable* A *has yielded a value* a_n*, the state of the system is the normalized eigenstate* $|a_n\rangle$

- *Postulate 6: The time evolution of a quantum system preserves the normalization of the associated ket. The time evolution of the state of a quantum system is described by* $|\psi(t)\rangle = \hat{U}(t,t_0)|\psi(t_0)\rangle$ *for some unitary operator* $\hat{U}$

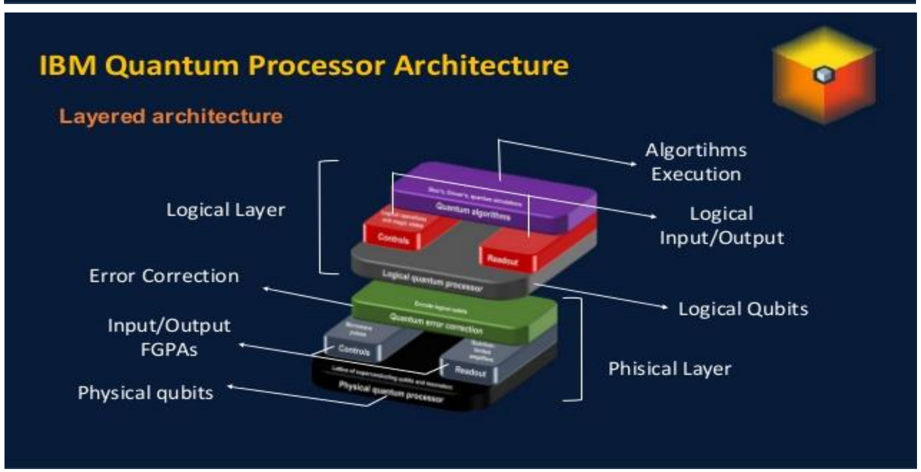
Qubit Processor Architectures

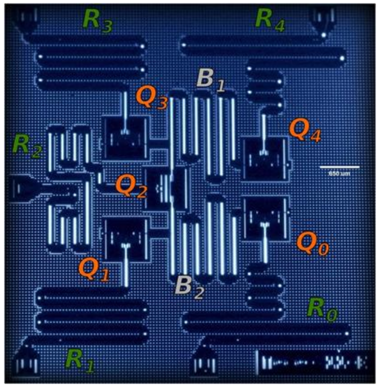
NORTHROP GRUMMAN



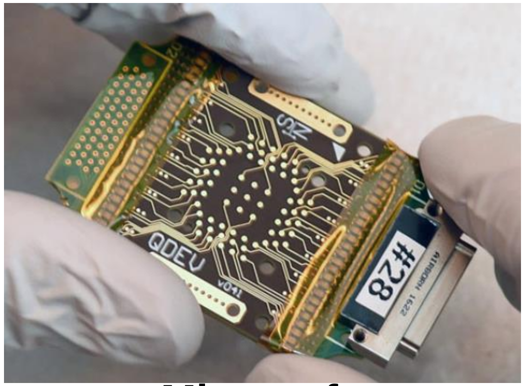


D-Wave Markets 1000 qubit computers for $10M - $15M



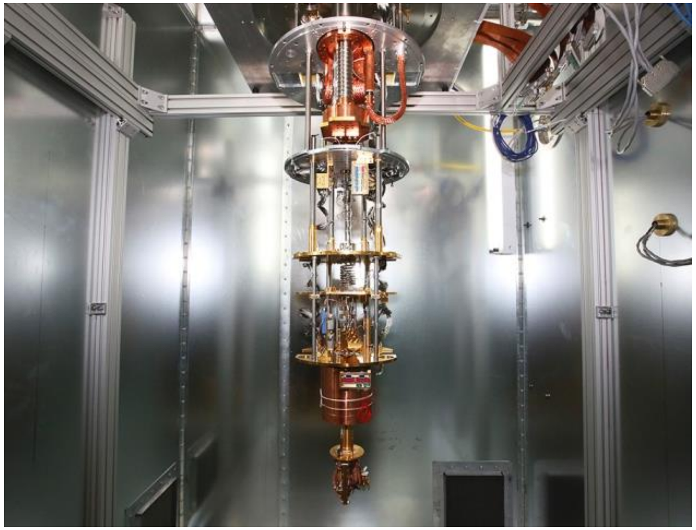


IBM 5 Qubit

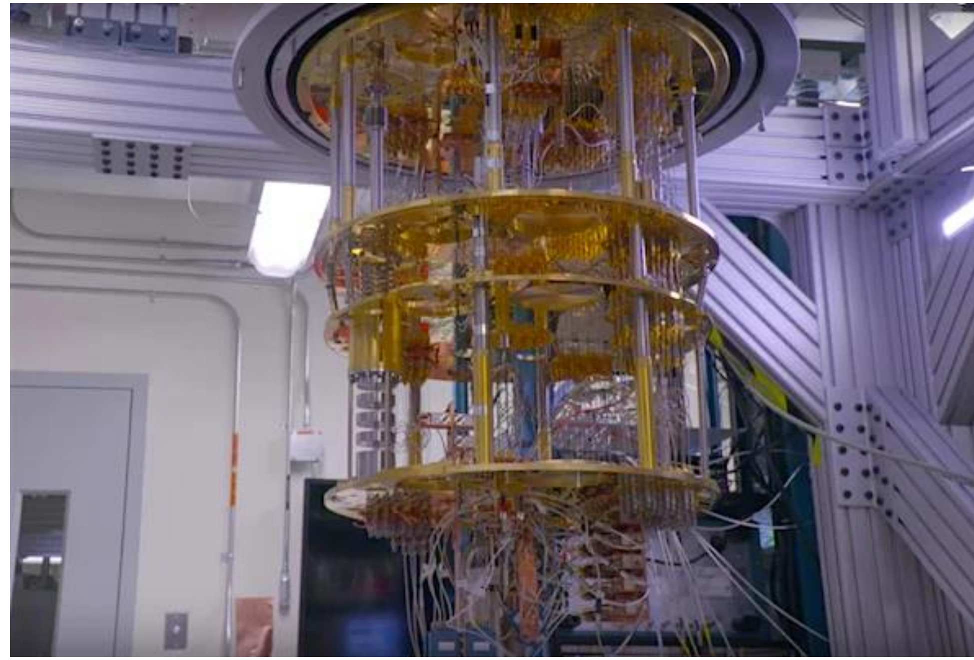


Microsoft

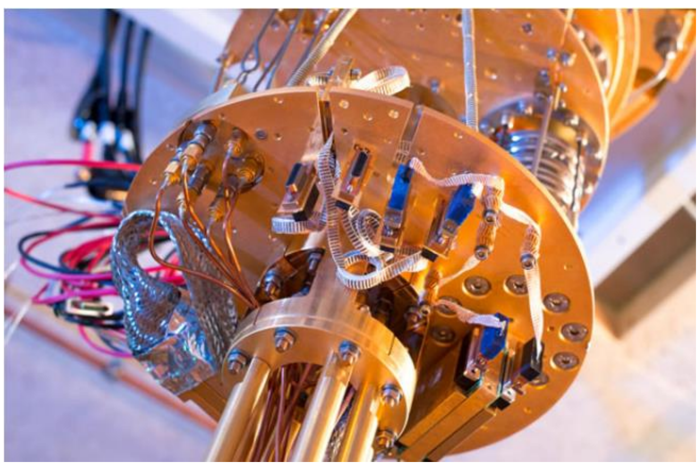
Physical Quantum Computer



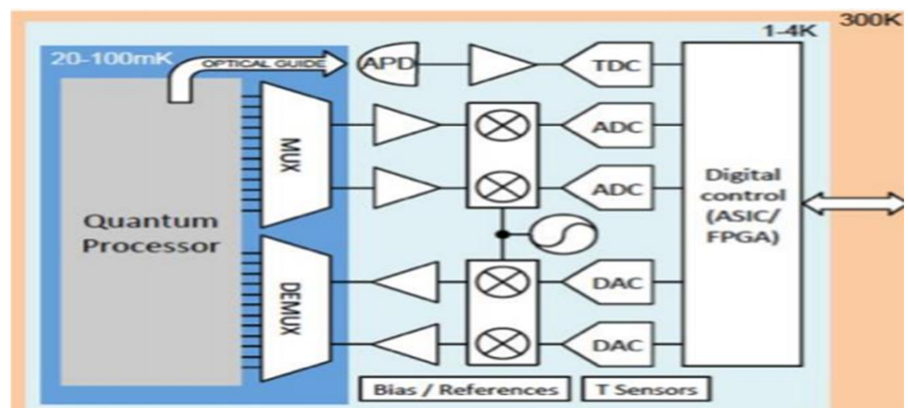
D-Wave



IBM



Microsoft



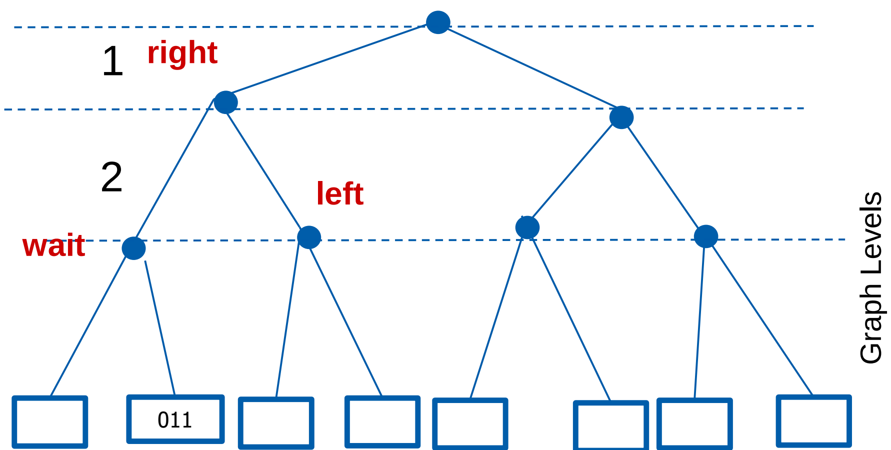
CMC Microelectronics Quantum & Classical Systems Integration

Quantum Random Access Memory (QRAM)

Quantum Register is an interface to an *addressable sequence of qubits*..

QRAM: In QRAM, the address and output registers are composed of qubits. The address register contains a superposition of addresses: $\sum_k b_k |k\rangle_a$ and the output registers post superposition of information correlated with the address register: $\sum_k b_k |k\rangle_a |D_k\rangle_d$

QRAM Model: "Bucket-brigade", architecture optimizes the retrieval of data to O(log 2^n) switches where "n" is the number of qubits in the address register. The basis of the architecture is to have qutrits instead of qubits allocated to the nodes of a bifurcation graph. "011" memory cell is an address register.



- **Quantum Entropy**: measure of information contained in a quantum system (von Neumann entropy):

$$S(\rho) = -tr(\rho \log_2 \rho) = -\sum_i \lambda_i \log_2 \lambda_i,$$

where λ_i are the members of the set of eigenvalues of ρ and $0 \log 0 \equiv 0$; $S(\rho)$ is nonnegative, maximum for mixed states For qubits $0 \le S(\rho) \le 1$; $S(\rho)$ provides information in measures of qubits

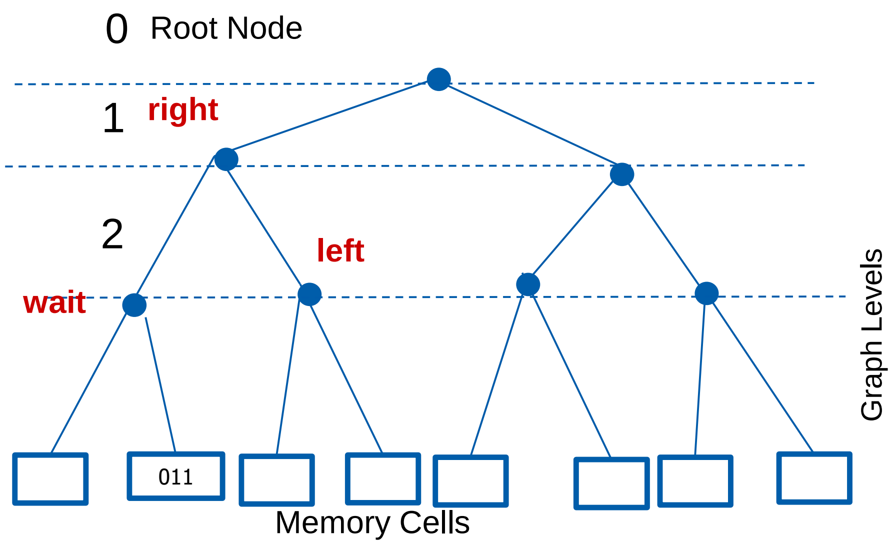
- **N qubits can store 2^N bits of information, e.g., DWave 1000 Qubits computer can store 2^{1000} ~ 1.07x10^{301} bits >> 10^{75} – 10^{82} atoms in the universe**

Note, however that N qubits can confer at most N bits of classical information

Quantum Random Access Memory (QRAM) (continued)

- |wait>, |left>, and |right> represent three-level qutrit quantum system. During each memory call the qutrit is in the |wait> state. The qubits of the address register are sent one by one through the graph and the wait state is transformed into |left> and |right> depending on the current qubit
- States not in |wait> states are routed immediately and the results are a superposition of routes
- The qutrit computation is to the O(1- €log N) where N is the number of qubits not in |wait> state

Bifurcation Graph



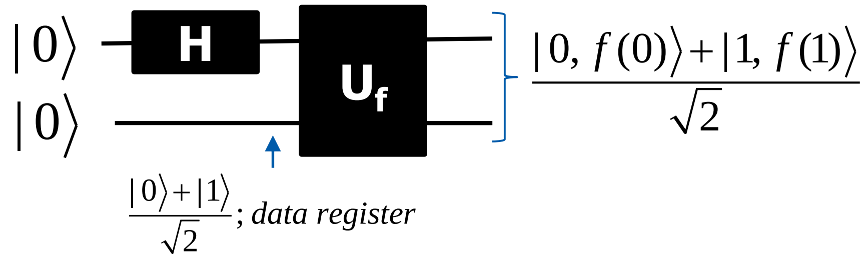
Quantum Circuits

- A quantum circuit consist of
  - Finite sequence of wires representing qubits or sequences of qubits (quantum registers)
  - Quantum gates that represent elementary operations from the particular set of operations implemented on a quantum machine
  - Measurement gates that represent a measurement operation, which is usually executed as the final step of a quantum algorithm
    - It is possible to perform the measurement on each qubit in canonical basis $\{|0\rangle,|1\rangle\}$ which corresponds to the measurement of a set of observables
  - Composite n-qubit circuit obey unitary evolution (every operation on multiple qubits is described by a unitary matrix)
  - Unitary implies reversibility: it establishes a bijective mapping between input and output bits (with the output and operations, the initial state can be recovered). Since all unitary operators **U** are invertible with $U^{-1}=U^{\dagger}$ we can always “un-compute" (reverse the computation) on a quantum computer

Quantum Parallelism

- Is there a *single operation* that evaluates a *single function* on at least two possible inputs to a quantum circuit without destroying superposition?
  - The results of such an operation is known as *Quantum Parallelism*

Simple example of Quantum Parallelism

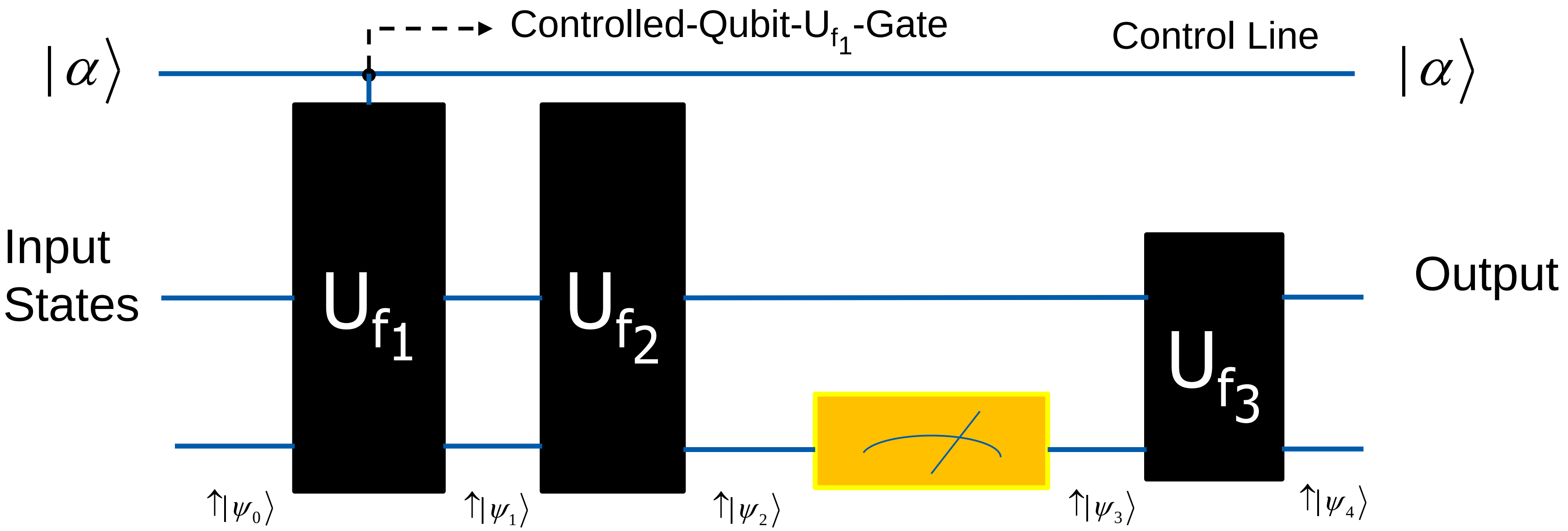


Function f in basis states: $\{0,1\} \mapsto \{0,1\}$
with appropriate sequence of quantum gates
$|\alpha, \beta\rangle$ transform to $|\alpha, \beta \oplus f(\alpha)\rangle$;
qubit α is called "data register"; qubit β is called "target register". If we apply a unitary transform U_f with β=0, such that the results becomes $|\alpha, f(\alpha)\rangle$
If we apply a Hadamard Gate on each data register it produces 2^n bits with n gates; then evaluate f with an appropriate U_f gate as in the example, we can generalize for n qubits with $|0\rangle^{\otimes n}|0\rangle$ the input state, Quantum Parallelism:

$$\frac{1}{\sqrt{2^n}}\sum_{\alpha}|\alpha\rangle|f(\alpha)\rangle$$

Quantum Circuits (continued)
Are one-shot circuits (run once from left to right)

NORTHROP GRUMMAN



- Circuit represents series of operations and measurements of n-qubit states
- Quantum gates U_{f_1} … U_{f_3} are operators that operate on qubits
- Each operator above is unitary and described by 2^n x 2^n matrix (*n* depends on input states)
- Each Line is an abstract wire connecting quantum logic gates (or series of gates)
- The meter symbol represents a measurement

Single Qubit Gates

NORTHROP GRUMMAN

1) Qubit NOT-Gate Representation: 2 x 2 matrix

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

Constraint:

$$U^{\dagger}U = I$$

$$(Identity\ \ matrix)$$

Input Amplitudes:

$$|\alpha|^2 + |\beta|^2 = 1$$

Output Amplitudes:

$$|\alpha^{'}|^2 + |\beta^{'}|^2 = 1$$

4) Qubit Pauli I-Gate Representation: 2 x 2 matrix

$$\sigma_0 = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$|0\rangle \rightarrow I \rightarrow |0\rangle$$

$$|1\rangle \rightarrow I \rightarrow |1\rangle$$

$$[\alpha \quad \beta]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \alpha\,|0\rangle + \beta\,|1\rangle$$

5) Qubit Pauli X-, Y-, and Z-Gates – Rotations about X, Y, and Z axis Representation:2 x 2 matrix

$$\sigma_X = X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\sigma_Y = Y = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$$

$$\sigma_Z = Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The unitary property provides other potential gates

Single Qubit Gates (continued)

4) Qubit Phase S-Gate

S

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} =$$

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

5) Qubit $\frac{\pi}{8}$ T-Gate

T

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} =$$

$$\alpha\,|0\rangle + e^{i\frac{\pi}{4}}\beta\,|1\rangle$$

Note: S = T^2

6) Qubit Hadamard H-Gate (square root NOT gate)

H

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

$$\alpha\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + \beta\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

7) Qubit Rotational R-Gates

R_X

$$\begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} = e^{-i\theta\frac{X}{2}}$$

R_Y

$$\begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} = e^{-i\theta\frac{Y}{2}}$$

R_Z

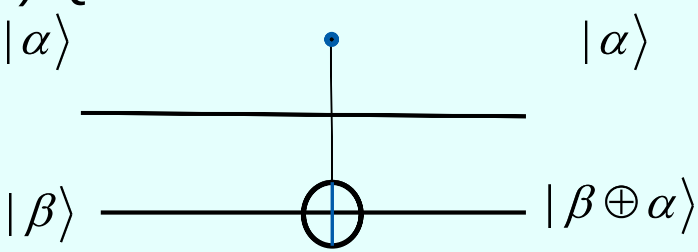
$$\begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & -e^{-i\frac{\theta}{2}} \end{bmatrix} = e^{-i\theta\frac{Z}{2}}$$

$$\sigma_{X,Y,Z}\ \textit{reduced form}:\ \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_{X,Y,Z};\ \textit{Identity and Pauli Operators}$$

Multi Qubit Gates

NORTHROP GRUMMAN

1) Qubit CNOT-Gate



- True quantum gates must be reversible. Reversibility require a control line which is unaffected by a unitary transformation. Implement by carrying the input with results
- $\oplus$ represent the classical XOR with input on the beta line and the control line in the alpha line
- The gate is a 2 qubit gate represented by a 4 x 4 matrix

$$|00\rangle \rightarrow CNOT \rightarrow |00\rangle;$$

$$|01\rangle \rightarrow CNOT \rightarrow |01\rangle;$$

$$|10\rangle \rightarrow CNOT \rightarrow |11\rangle;$$

$$|11\rangle \rightarrow CNOT \rightarrow |10\rangle$$

$$\left(\alpha\,|0\rangle + \beta\,|1\rangle\right)|1\rangle \rightarrow CNOT \rightarrow \alpha\,|01\rangle + \beta\,|10\rangle;$$

$$|0\rangle\left(\alpha\,|0\rangle + \beta\,|1\rangle\right) \rightarrow CNOT \rightarrow \alpha\,|00\rangle + \beta\,|01\rangle;$$

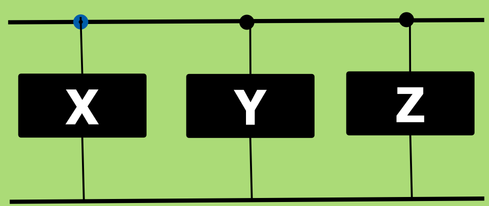
$$|1\rangle\left(\alpha\,|0\rangle + \beta\,|1\rangle\right) \rightarrow CNOT \rightarrow \alpha\,|11\rangle + \beta\,|10\rangle;$$

$$\begin{pmatrix} 1000 \\ 0100 \\ 0001 \\ 0010 \end{pmatrix}\begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix} = \alpha\,|00\rangle + \beta\,|11\rangle$$

2-Qubit CNOT-Gate treatment in Backup Slides

Multi Qubit Gates (continued)

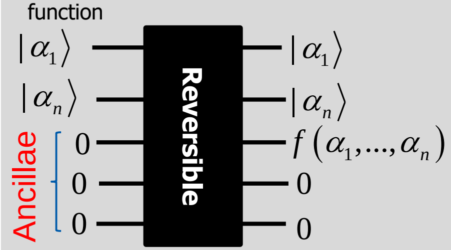
2) Controlled X, Y, Z Gates



- CNOT is a controlled-X-gate
- $SXS^{\dagger}$ = controlled-Y-gate
- HXH = controlled-Z-gate

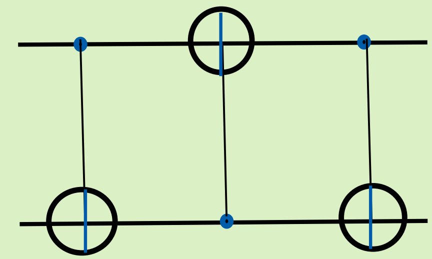
3) Reversible Circuit

At end of computation all ancillae retain initial values, except one ancilla bit, designated as the "answer" bit, carries the value of the function



Ancilla bit is a storage/garbage bit

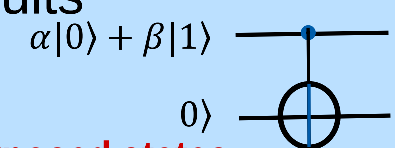
4) Swap Qubit States



SWAP12=CNOT12→CNOT21→CNOT12

$$(\alpha,\beta) \to (\alpha, \alpha\oplus\beta) \to (\beta, \alpha\oplus\beta) \to (\beta,\alpha)$$

5) Copying Circuits



Only on non-superposed states

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha|00\rangle + \beta|10\rangle; \text{ combined state}$$

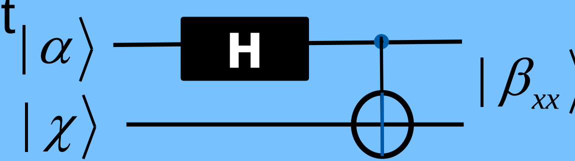
$$(\alpha|00\rangle + \beta|10\rangle) \to CNOT \to \alpha|00\rangle + \beta|11\rangle;$$

not a copy of original state

$$(\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle) \neq \alpha|00\rangle + \beta|11\rangle;$$

A qubit in an input unknown state cannot be copied. It must be measured before being copied. The information held in the probability amplitudes α and β is lost.

6) Bell State Circuit



Entangled states are produced: β_{00}, β_{01}, β_{10}, and β_{11}

$$|00\rangle \to \beta \to \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \to \beta_{00};$$

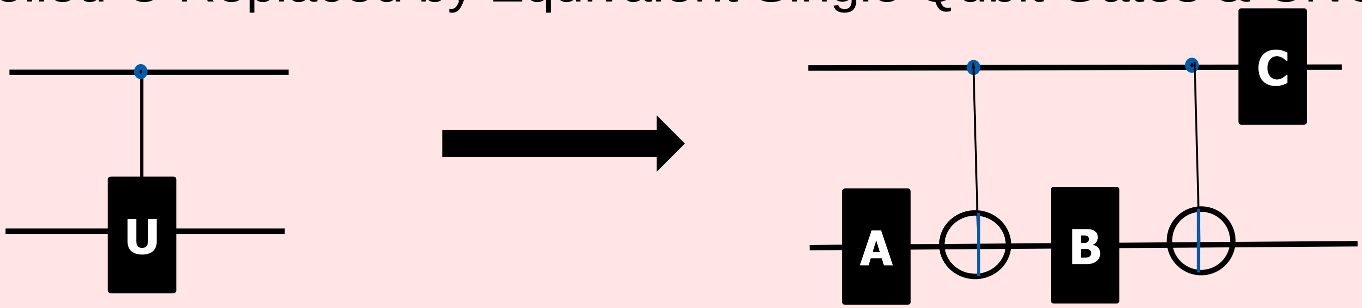
$$|01\rangle \to \beta \to \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \to \beta_{01};$$

$$|10\rangle \to \beta \to \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \to \beta_{10};$$

$$|11\rangle \to \beta \to \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \to \beta_{11};$$

Equivalent Quantum Gate Operations (some examples)

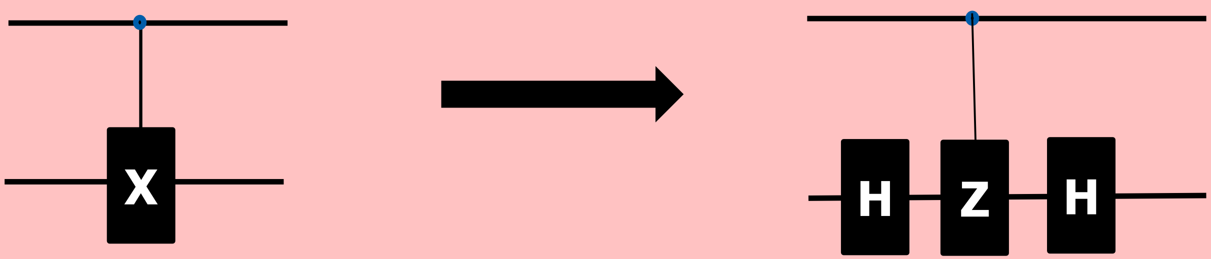
7) Controlled-U Replaced by Equivalent Single Qubit Gates & CNOT gate



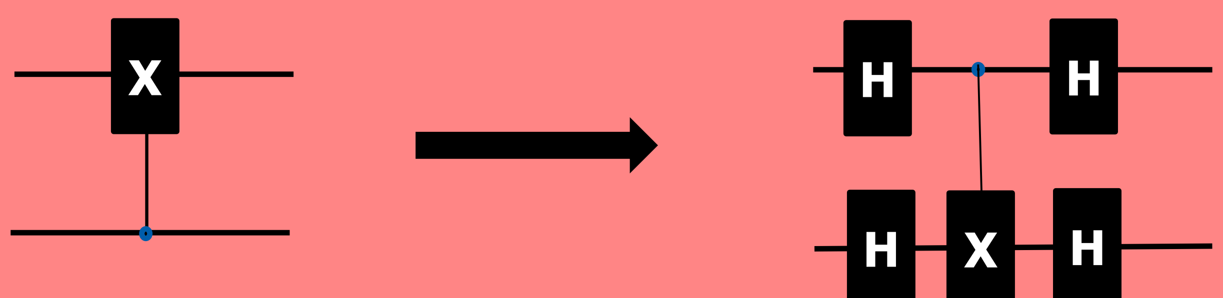
Controlled-Unitary Gate

Single Qubit Gates: A, B, C

8) Controlled-Pauli X Gate Replaced by Hadamard and Controlled-Pauli Z Gate



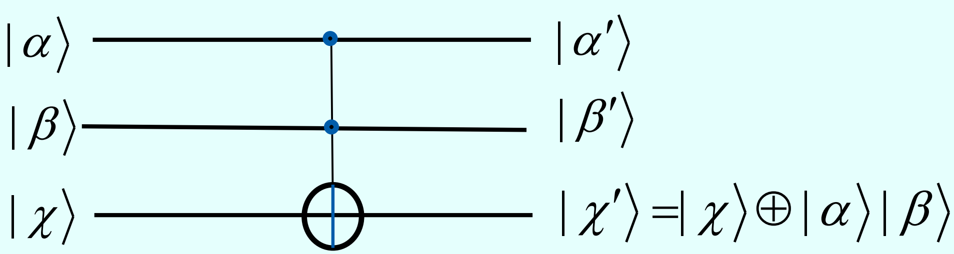
9) Controlled-Pauli X Gate Equivalent Circuit



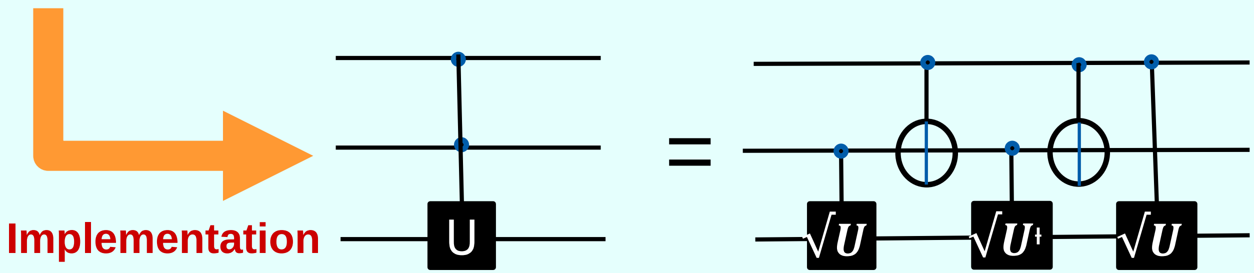
Multi Qubit Gates (continued)

NORTHROP GRUMMAN

10) Qubit Toffoli Controlled-CNOT (CCNOT) or Deutsch ($\pi/2$) Gate



- Universal reversible gate
- Fast, stable to imperfections, and has high fidelity for fault-tolerant quantum computation
- Control qubits remain unaffected
- Third target qubit is flipped if both control lines are set to 1, else it is left alone.

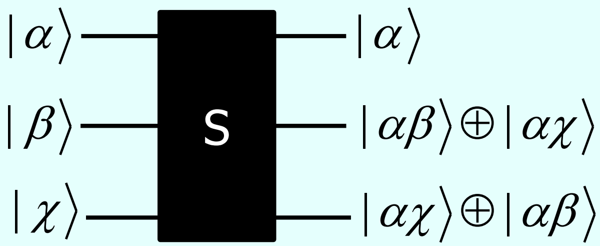


$$\begin{aligned} &\textit{Toffoli Matrix}: \\ &= |000\rangle\langle 000| + |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| + \\ &\quad |100\rangle\langle 100| + |101\rangle\langle 101| + |110\rangle\langle 111| + |111\rangle\langle 110| \end{aligned} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Permutations in 8 Dimension Hilbert Space that swaps the last two entries

Multi Qubit Gates (continued)

11) Qubit Fredkin (Controlled-SWAP) Gate



- Universal reversible gate
- Factor impossibly large number in short time periods
- Secure quantum communications - direct comparison of two sets of qubits for equality i.e., the two digital signatures are the same

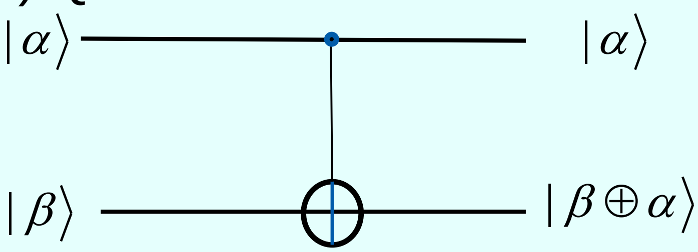
$$\begin{aligned} &\textit{Fredkin Matrix}: \\ &= |000\rangle\langle 000| + |001\rangle\langle 001| + |010\rangle\langle 010| + |011\rangle\langle 011| + \\ &\quad |100\rangle\langle 100| + \underbrace{|101\rangle\langle 110| + |110\rangle\langle 101|} + |111\rangle\langle 111| \end{aligned} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Permutations in 8 Dimension Hilbert Space that swaps the 101 and 110

Multi Qubit Gates

NORTHROP GRUMMAN

1) Qubit CNOT-Gate



- True quantum gates must be reversible. Reversibility require a control line which is unaffected by unitary transformation. Implement by carrying the input with results
- $\oplus$ represent the classical XOR with input on the beta line and the control line in the alpha line
- The gate is a 2 qubit gate represented by a 4 x 4 matrix

$$|00\rangle \rightarrow CNOT \rightarrow |00\rangle;$$
$$|01\rangle \rightarrow CNOT \rightarrow |01\rangle;$$
$$|10\rangle \rightarrow CNOT \rightarrow |11\rangle;$$
$$|11\rangle \rightarrow CNOT \rightarrow |10\rangle$$

$$(\alpha|0\rangle + \beta|1\rangle)|1\rangle \rightarrow CNOT \rightarrow \alpha|01\rangle + \beta|10\rangle;$$
$$|0\rangle(\alpha|0\rangle + \beta|1\rangle) \rightarrow CNOT \rightarrow \alpha|00\rangle + \beta|01\rangle;$$
$$|1\rangle(\alpha|0\rangle + \beta|1\rangle) \rightarrow CNOT \rightarrow \alpha|11\rangle + \beta|10\rangle;$$

$$\begin{pmatrix} 1000 \\ 0100 \\ 0001 \\ 0010 \end{pmatrix}\begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{pmatrix} = \alpha|00\rangle + \beta|11\rangle$$

2) Qubit NOT *Two* Gates Which Acts On Qubit 2

$$NOT_2 = I \otimes X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & 0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ 0\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & 1\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix};$$

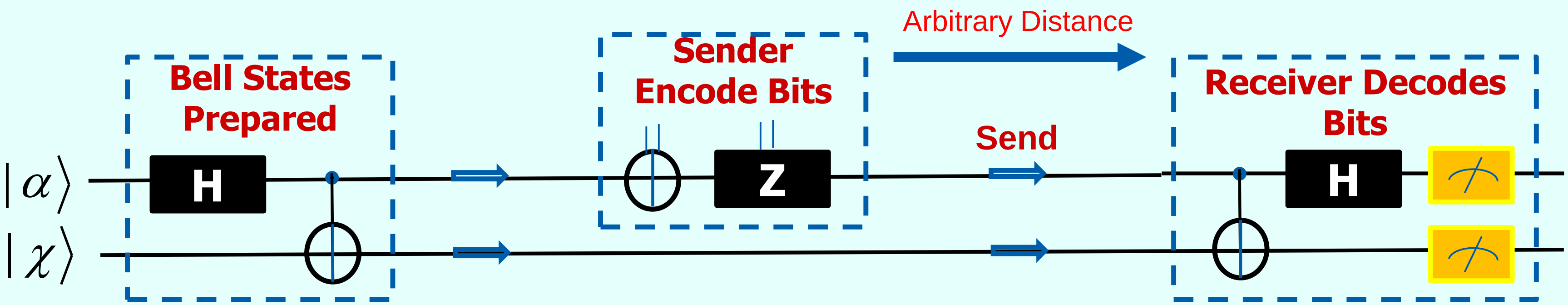
$$\begin{pmatrix} 0100 \\ 1000 \\ 0001 \\ 0010 \end{pmatrix};$$

$$|00\rangle \rightarrow NOT_2 = I \otimes X \rightarrow |01\rangle;$$
$$|01\rangle \rightarrow NOT_2 = I \otimes X \rightarrow |00\rangle;$$
$$|10\rangle \rightarrow NOT_2 = I \otimes X \rightarrow |11\rangle;$$
$$|11\rangle \rightarrow NOT_2 = I \otimes X \rightarrow |10\rangle;$$

$$|00\rangle \rightarrow I \otimes X \rightarrow |01\rangle;$$
$$|01\rangle \rightarrow I \otimes X \rightarrow |00\rangle;$$
$$|10\rangle \rightarrow I \otimes X \rightarrow |11\rangle;$$
$$|11\rangle \rightarrow I \otimes X \rightarrow |10\rangle$$

13) Qubit Superdense Coding

Superdense coding takes a quantum state to two classical bits. It is a method for building shared quantum entanglement in order to increase the rate at which information may be sent through a noiseless quantum channel. Sending a single qubit noiselessly between sender and receiver gives maximum communication rate of one bit per qubit. If the sender's qubit is maximally entangled with a qubit in the receiver's possession, then dense coding increases the maximum rate to two bits per qubit.



14) Qubit Error Correction Circuit

$|\psi_1\rangle = \alpha|001\rangle + \beta|110\rangle;$

$|\psi_2\rangle = \alpha|00100\rangle + \beta|11000\rangle;$

$|\psi_3\rangle = \alpha|00101\rangle + \beta|11001\rangle;$

$|\psi_4\rangle = \left(\alpha|001\rangle + \beta|110\rangle\right)\otimes|0\rangle|1\rangle;$

M_1 and M_2 read 01 on lines 4 and 5. Feed 01 (error syndrome) into the QEC which performs operations in the table below.

Apply qubit flip to line 3:

$|\psi_5\rangle = \alpha|000\rangle + \beta|111\rangle$

Errors in qubit superposition and entanglement occur due to increase in thermal motion of qubits as a result of environmental temperature increase. Qubit encoding errors are also possible. Reasons for single qubit errors:

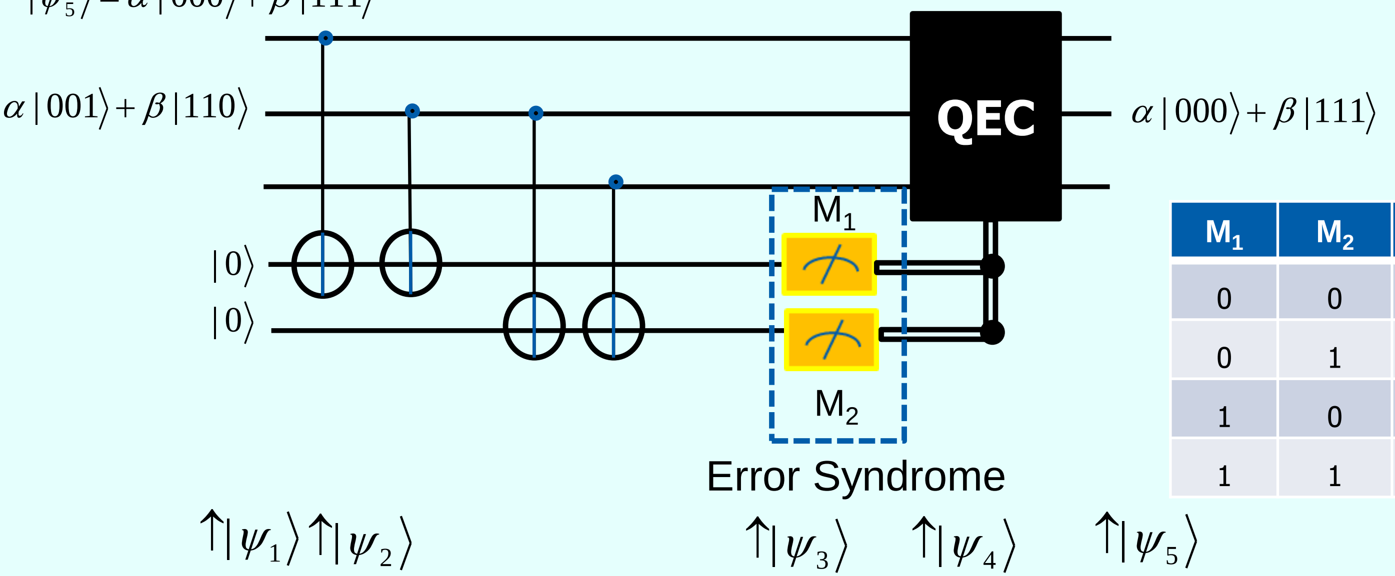
1) Qubit Flip X: $X|0\rangle = |0\rangle; X|1\rangle = |0\rangle$
2) Qubit Phase Flip Z: $Z|0\rangle = |0\rangle; Z|1\rangle = -|1\rangle$
3) Qubit Complete Decoherence ρ:

$$\rho \to \frac{1}{2}\left(\rho + Z\rho Z^\dagger\right);$$

where $\rho = \sum O_i \rho O_i^\dagger$; O is 2x2 matrix

4) Qubit Rotation R_θ: $R_\theta|0\rangle = |0\rangle; R_\theta|1\rangle = e^{i\theta}|1\rangle$
5) Basis states: {|0>, |1>}

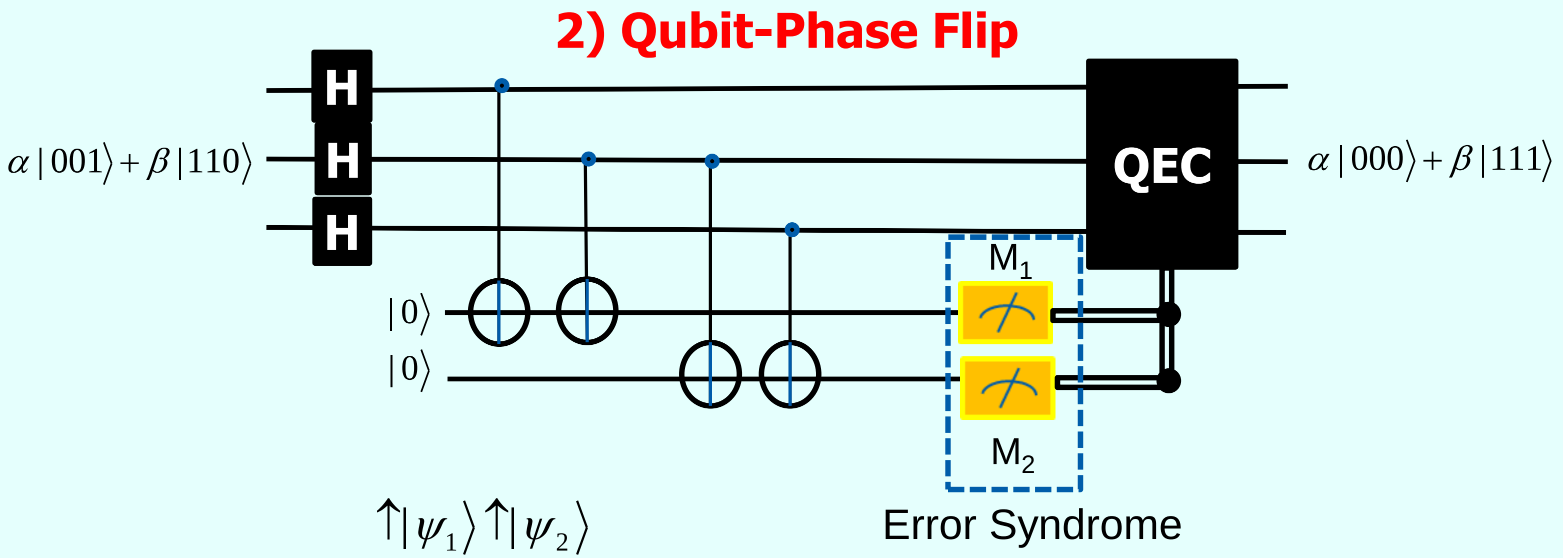
1) Qubit-Flip (Amplitude Flip)



M_1	M_2	Action
0	0	No action |111>→|111>
0	1	Flip qubit 3; |110>→|111>
1	0	Flip qubit 2; |101>→|111>
1	1	Flip qubit 1; |011>→|111>

Multi Qubit Gates (continued)

15) Qubit Error Correction Circuit



- Same circuit as the amplitude flip circuit, except the Hadamard gates are added to the first three lines. Repetition code in the Hadamard gates correct for phase errors.
- Errors happen between the encoding and the circuit
- Suppose the input state is: $|\psi_1\rangle=\alpha|++-\rangle+\beta|--+\rangle$ and phase flip occurs in line 2:

  $|\psi_2\rangle=\left(\alpha|001\rangle+\beta|110\rangle\right)|00\rangle$; note that is the same as in the qubit-flip (amplitude flip)
- Since the rest of the circuit is the same as the qubit-flip case. The output of QEC is: $\alpha|000\rangle+\beta|111\rangle$

Theorem: If a quantum error correcting code (QECC) corrects error A and B, then it also corrects errors $\alpha A+\beta B$

NORTHROP GRUMMAN

16) Qubit Error Correction Circuit

3) Qubit-Decoherence

Decoherence is the loss of coherence in a quantum system due to interactions with external environment.

Decoherence in qubit system can be modeled by introducing a relative phase:

$$|0\rangle \to |0\rangle \ and\ |1\rangle \to e^{i\theta}|1\rangle,\ i.e.,$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \to \alpha|0\rangle + e^{i\theta}\beta|1\rangle;$$

$$i.e.,|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}};$$

$$\rho = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \to \frac{1}{2}\begin{pmatrix} 1 & e^{-i\theta} \\ e^{i\theta} & 1 \end{pmatrix}$$

Density Operator for state $|\psi\rangle$:

$$\rho = |\psi\rangle\langle\psi|;$$

Time dependent Density Operator:

$$\rho(t) = U\rho(t_0)U^{\dagger};\ \text{U is Unitary matrix}$$

$$\rho^2 = (|\psi\rangle\langle\psi|)(|\psi\rangle\langle\psi|) = |\psi\rangle(\langle\psi|\psi\rangle)\langle\psi| = |\psi\rangle\langle\psi| = \rho;$$

$$Tr(\rho^2) = 1$$

A global phase multiplies all superpositions, whereas a relative phase multiplies only a single term in the superposition and does not change measurements. We map, instead to a decoherent free subspace using logical gates in order avoid problems with physical global and relative phases:

$$|0_L\rangle = \frac{|0\rangle|1\rangle - i|1\rangle|0\rangle}{\sqrt{2}};|1_L\rangle\frac{|0\rangle|1\rangle + i|1\rangle|0\rangle}{\sqrt{2}}$$

Introduce collective dephasing:

$$|0_L\rangle = \frac{|0\rangle e^{i\theta}|1\rangle - ie^{i\theta}|1\rangle|0\rangle}{\sqrt{2}} = e^{i\theta}|0_L\rangle;$$

$$|1_L\rangle\frac{|0\rangle e^{i\theta}|1\rangle + ie^{i\theta}|1\rangle|0\rangle}{\sqrt{2}} = e^{i\theta}|1_L\rangle;$$

Each logical qubit has been altered by an overall global phase $e^{i\theta}$ and an arbitrary logical qubit is unchanged by decoherence. Hence error correction has been applied:

$$\boxed{|\psi_L\rangle = \alpha|0_L\rangle + \beta_L|1\rangle \to e^{i\theta}\alpha|0_L\rangle + e^{i\theta}\beta_L|1\rangle = e^{i\theta}|\psi_L\rangle}$$

NORTHROP GRUMMAN

17) Qubit Error Correction Circuit

3) Qubit-Continuous rotational error

R_θ acts on the j^{th} qubit

$$R_\theta^j |\psi\rangle = \cos\frac{\theta}{2}|\psi\rangle - i\sin\frac{\theta}{2}Z^j|\psi\rangle$$

$$\Rightarrow \cos\frac{\theta}{2}|\psi\rangle|I\rangle - i\sin\frac{\theta}{2}Z^j|\psi\rangle|Z^j\rangle$$

Error Syndrome

Error syndrome is formed by measuring enough operators to determine the location error

Measuring the error syndrome collapses the state:

Probability:

$$\cos^2\frac{\theta}{2}: \quad |\psi\rangle \qquad (\textit{no correction needed})$$

$$\sin^2\frac{\theta}{2}: \quad Z^j|\psi\rangle \quad (\textit{Corrected with } Z^j)$$

Pauli Group Stabilizers
9-Qubit Error Syndrome Code

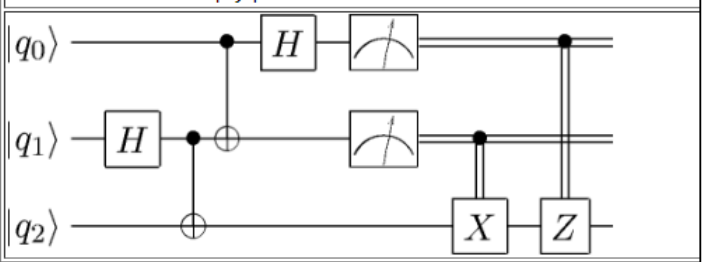
	Operators for Error Syndrome								
M_1	Z	Z							
M_2		Z	Z						
M_3				Z	Z				
M_4					Z	Z			
M_5							Z	Z	
M_6								Z	Z
M_7	X	X	X	X	X	X			
M_8				X	X	X	X	X	X

These generate a group, the stabilizer of the code with all ***M Pauli operators*** with property: $M|\psi\rangle = |\psi\rangle$ and all encoded sates $|\psi\rangle$

QASM2CIRC - MIT
Simple Quantum Teleportation Circuit

https://www.media.mit.edu/quanta/qasm2circ/

```
#
# File:   test2.qasm
# Date:   29-Mar-04
# Author: I. Chuang
#
# Sample qasm input file - simple teleportation circuit
#
        qubit   q0
        qubit   q1
        qubit   q2

        h       q1      # create EPR pair
        cnot    q1,q2
        cnot    q0,q1   # Bell basis measurement
        h       q0
        nop     q1
        measure q0
        measure q1
        c-x     q1,q2   # correction step
        c-z     q0,q2
```




```
%  Time 01:
%    Gate 00 h(q1)
%  Time 02:
%    Gate 01 cnot(q1,q2)
%  Time 03:
%    Gate 02 cnot(q0,q1)
%  Time 04:
%    Gate 03 h(q0)
%    Gate 04 nop(q1)
%  Time 05:
%    Gate 05 measure(q0)
%    Gate 06 measure(q1)
%  Time 06:
%    Gate 07 c-x(q1,q2)
%  Time 07:
%    Gate 08 c-z(q0,q2)
% Qubit circuit matrix:
%
% q0: n  , n  , gCA, gDA, gEA, N  , gGA, N
% q1: gAB, gBB, gCB, gDB, gEB, gFB, N  , N
% q2: n  , gBC, n  , n  , n  , gFC, gGC, n
\documentclass[11pt]{article}
\input{xyqcirc.tex}
% definitions for the circuit elements
\def\gAB{\op{H}\w\A{gAB}}
\def\gBB{\b\w\A{gBB}}
\def\gBC{\o\w\A{gBC}}
\def\gCA{\b\w\A{gCA}}
\def\gCB{\o\w\A{gCB}}
\def\gDA{\op{H}\w\A{gDA}}
\def\gDB{*-{}\w\A{gDB}}
\def\gEA{\meter\w\A{gEA}}
\def\gEB{\meter\w\A{gEB}}
\def\gFB{\b\W\A{gFB}}
\def\gFC{\op{X}\w\A{gFC}}
\def\gGA{\b\W\A{gGA}}
\def\gGC{\op{Z}\w\A{gGC}}
```

```
% definitions for bit labels and initial states

\def\bA{ \q{q_{0}}}
\def\bB{ \q{q_{1}}}
\def\bC{ \q{q_{2}}}

% The quantum circuit as an xymatrix

\xymatrix@R=5pt@C=10pt{
    \bA & \n   &\n   &\gCA &\gDA &\gEA
&\N   &\gGA &\N
\\  \bB & \gAB &\gBB &\gCB &\gDB
&\gEB &\gFB &\N   &\N
\\  \bC & \n   &\gBC &\n   &\n   &\n
&\gFC &\gGC &\n
%
% Vertical lines and other post-xymatrix latex
%
\ar@{-}"gBC";"gBB"
\ar@{-}"gCB";"gCA"
\ar@{=}"gFC";"gFB"
\ar@{=}"gGC";"gGA"
}

\end{document}
```

CodeProject Quantum Java Code

```
/**
* Constructs a new <code>Qubit</code> object.
* @param no0 complex number
* @param no1 complex number
*
*/
public Qubit(ComplexNumber no0, ComplexNumber no1) {
qubitVector = new ComplexNumber[2];
qubitVector[0] = no0;
qubitVector[1] = no1;
}

/**
* Constructs a new <code>Qubit</code> object.
* @param qubitVector an array of 2 complex numbers
*/
public Qubit(ComplexNumber[] qubitVector) {
this.qubitVector=Arrays.copyOf(qubitVector, qubitVector.length);
}

/**
* Return the qubit represented as an array of 2 complex numbers.
* @return qubit
*/
public ComplexNumber[] getQubit() {
ComplexNumber[] copyOfQubitVector = qubitVector;
                                        return copyOfQubitVector;
}
```

```
/**
 * Check if qubit state is valid
 * @return true if the state is valid, otherwise false
 */
public boolean isValid(){
double sum=0.0;
for(ComplexNumber c:this.qubitVector){
double mod=ComplexMath.mod(c);
sum+=mod*mod;
}
return (sum==1.0);
}

public class QubitZero extends Qubit {
// Construct a new <code> QubitZero</code> object.
public QubitZero() {
super(new ComplexNumber(1.0, 0.0), new ComplexNumber(0.0, 0.0));
}
}
/**
 * Currently Implemented Quantum Gates.
*/
public enum EGateTypes {
// Hadamard Gate
E_HadamardGate,
// Pauli-X Gate
E_XGate,
// Pauli-Z Gate
E_ZGate,
// CNOT Gate
E_CNotGate
}
```

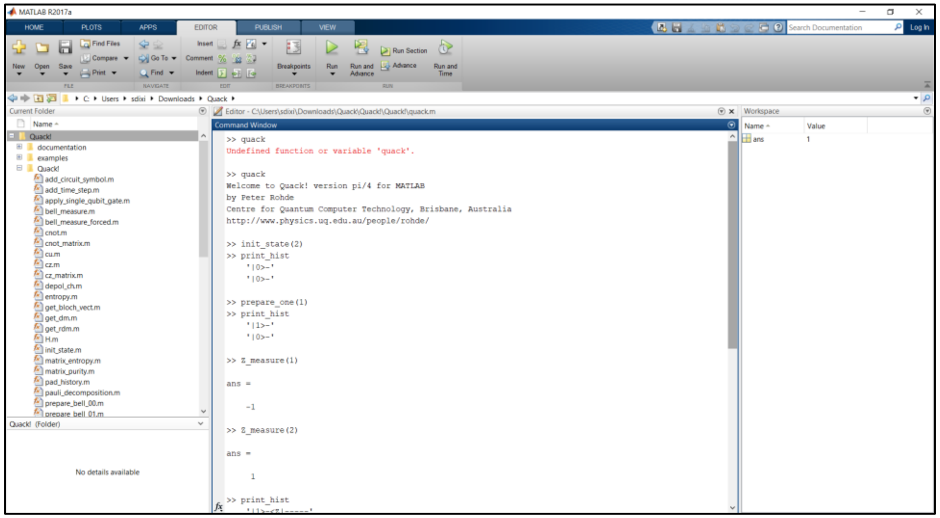
QISKit SDK – Quantum Python Code Example

https://qiskit.org/documentation/quickstart.html

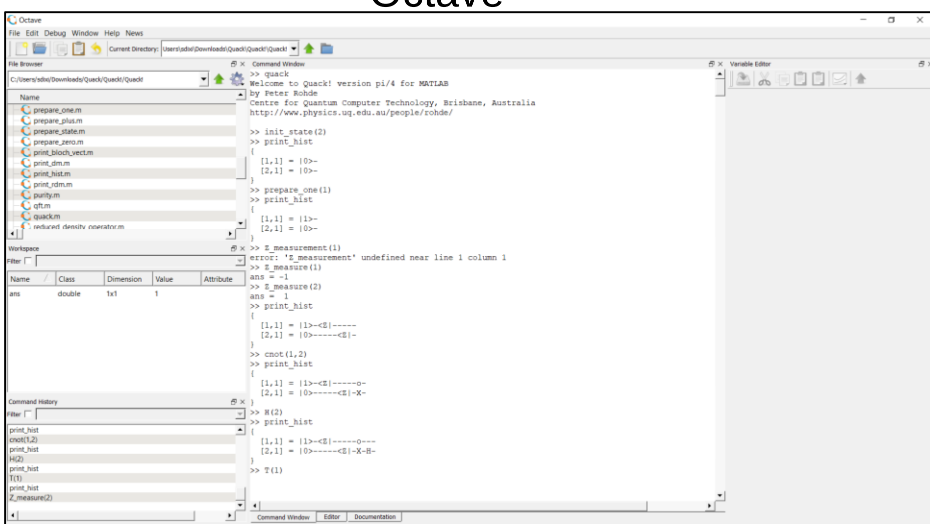
```
# Import the QISKit SDK
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import available_backends, execute

# Create a Quantum Register with 2 qubits.
q = QuantumRegister(2)
# Create a Classical Register with 2 bits.
c = ClassicalRegister(2)
# Create a Quantum Circuit
qc = QuantumCircuit(q, c)

# Add a H gate on qubit 0, putting this qubit in superposition.
qc.h(q[0])
# Add a CX (CNOT) gate on control qubit 0 and target qubit 1, putting
# the qubits in a Bell state.
qc.cx(q[0], q[1])
# Add a Measure gate to see the state.
qc.measure(q, c)

# See a list of available local simulators
print("Local backends: ", available_backends({'local': True}))

# Compile and run the Quantum circuit on a simulator backend
job_sim = execute(qc, "local_qasm_simulator")
sim_result = job_sim.result()

# Show the results
print("simulation: ", sim_result)
print(sim_result.get_counts(qc))
```

QUACK Simulator In MATLAB/OCTAVE

Matlab



Octave



```
>> quack
Welcome to Quack! version pi/4 for MATLAB
by Peter Rohde
Centre for Quantum Computer Technology, Brisbane, Australia
http://www.physics.uq.edu.au/people/rohde/
>> init_state(2)
>> print_hist
{
  [1,1] = |0>-
  [2,1] = |0>-
}
```

Initialize 2-qubit register to ground state

```
>> prepare_one(1)
>> print_hist
{
  [1,1] = |1>-
  [2,1] = |0>-
}
```

Initialize states to |1> and |0>

```
>> Z_measure(1)
ans = -1
>> Z_measure(2)
ans =  1
```

Measure spins of |1> and |0> along the z-axis (-1 => spin down)

```
>> print_hist
{
  [1,1] = |1>-<Z|-----
  [2,1] = |0>-----<Z|-
}
```

Note the entry points in the circuit are shown on the right side

```
>> cnot(1,2)
>> print_hist
{
  [1,1] = |1>-<Z|-----o-
  [2,1] = |0>-----<Z|-X-
}
```

Apply CNOT with first qubit as control

```
>> H(2)
>> print_hist
{
  [1,1] = |1>-<Z|-----o---
  [2,1] = |0>-----<Z|-X-H-
}
```

Now apply Hadamard on second qubit

```
>> T(1)
>> print_hist
{
  [1,1] = |1>-<Z|-----o---T-
  [2,1] = |0>-----<Z|-X-H---
}
```

Apply phase shift T gate to first qubit

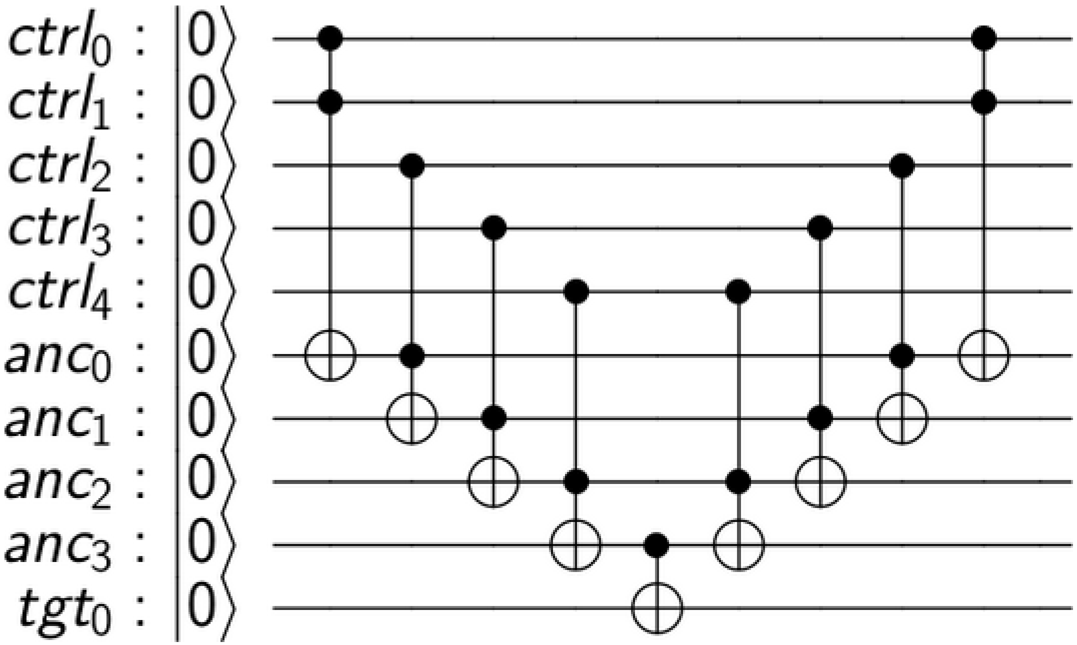
```
>> Z_measure(2)
ans =  1
>>
```

Measure spin of second qubit along the z-axis

http://www.peterrohde.org/media/software/

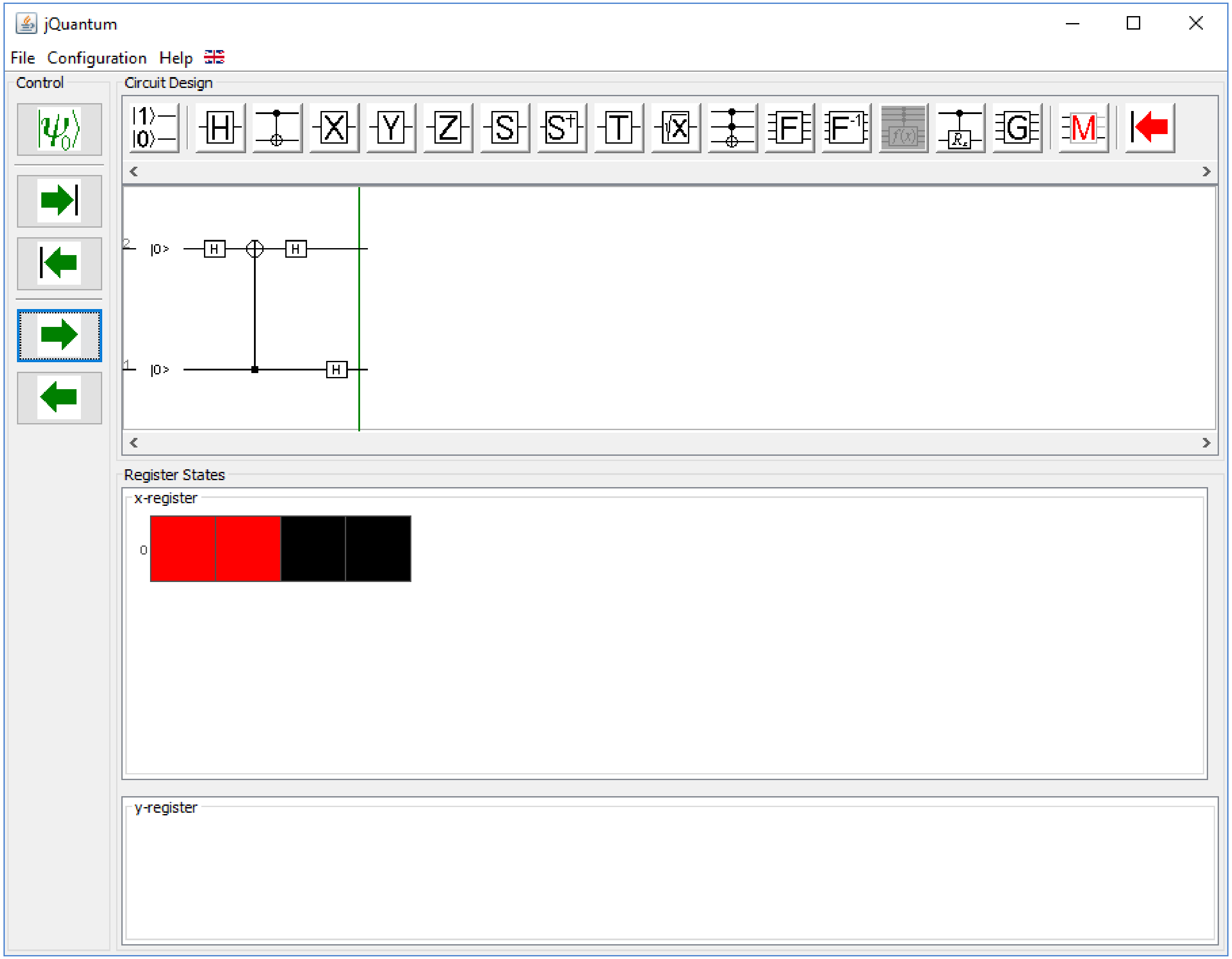
5 Qubit Tofolli Gate and QISKIT Programming

```
from qiskit import QuantumRegister, QuantumCircuit

n = 5  # must be >= 2

ctrl = QuantumRegister(n, 'ctrl')
anc = QuantumRegister(n-1, 'anc')
tgt = QuantumRegister(1, 'tgt')

circ = QuantumCircuit(ctrl, anc, tgt)

# compute
circ.ccx(ctrl[0], ctrl[1], anc[0])
for i in range(2, n):
    circ.ccx(ctrl[i], anc[i-2], anc[i-1])

# copy
circ.cx(anc[n-2], tgt[0])

# uncompute
for i in range(n-1, 1, -1):
    circ.ccx(ctrl[i], anc[i-2], anc[i-1])
circ.ccx(ctrl[0], ctrl[1], anc[0])

from qiskit.tools.visualization import circuit_drawer
circuit_drawer(circ)
```



https://qiskit.org/documentation/qiskit.html

JQuantum Java Quantum Simulator



http://jquantum.sourceforge.net/

Quantum Algorithms

- Quantum algorithms are realized by quantum circuits
  - Complexity optimization
- Turing machine complexity definitions
  - **P** is the set of problems that can be solved by deterministic Turing machines in **P**olynomial number of steps
  - **NP** is the set of problems that can be solved by **N**ondeterministic Turing machines in **P**olynomial number of steps

$$P \subseteq NP; P = NP?\,(not\;\; proven\;\; yet)$$

  - **coP** is the set of problems whose **co**mplements can be solved by deterministic Turing machine in **P**olynomial number of steps
  - **coNP** is the set of problems whose **co**mplements can be solved by a **N**ondeterministic Turing machine in **P**olynomial number of steps

$$NP \subseteq PSPACE$$

  - **PSPACE** is the set of problems that can be solved by deterministic Turing machine using a Polynomial number of SPACEs on the tape

$$P \subseteq coP \subseteq coNP \;\; ; \; coNP \subseteq PSPACE$$

- Probabilistic Turing machine (PTM) complexity definitions
  - **BPP** is the set of problems that can be solved by **P**robabilistic Turing machines in **P**olynomial time with some errors possible

Turing Machine "String-101" Execution Time

	Exact	Probable
Deterministic	N +N/2	NA
Probabilistic	N +N/2	N/2
Quantum	N/2	NA

  - **RP** is the set of problems that can be solved by **P**robabilistic Turing machines in **P**olynomial time with false negatives possible
  - **coRP** replaces "false negatives" with "false positives" in **RP** definition
  - **ZPP** replaces "some errors possible" with "zero error" in **BPP** definition

- Quantum Turing machine (QTM) complexity definitions

– **BQP, ZQP,**

– Is a set of problems that can be solved by QTM in **P**olynomial time with **B**ounded error on both sides

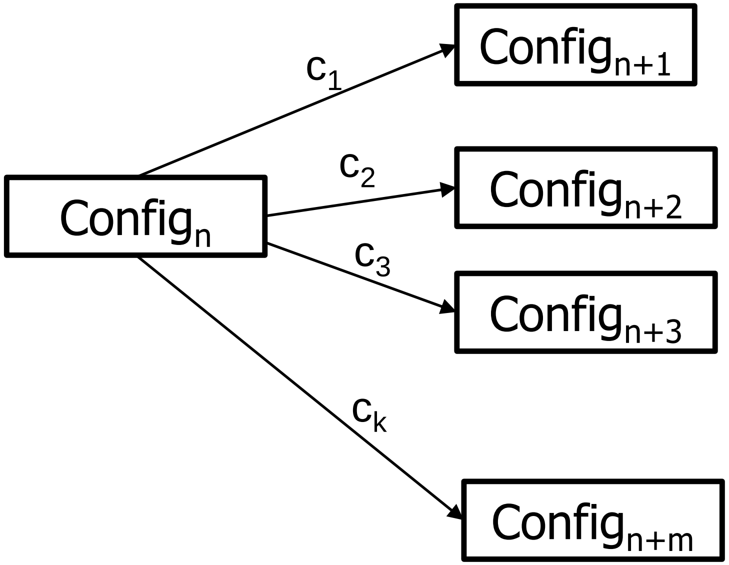
– **EQP**

- Replaces **B**ounded error with "**E**xactly (without error)" in definition of QTM

– **QSPACE** $QSPACE(f(n)) \subseteq SPACE((f(n))^2)$

Quantum Computing Algorithms

- Quantum Turing Machine (QTM)
  - Is well formed if the constructed U_M preserves isometric inner product in □ complex space
    - QTM is similar to the probabilistic Turning machine (PTM), except that the probability amplitudes are complex number amplitudes
    - Probabilistic TM (PTM) traverses the tape left to right; QTM traverses in both directions simultaneously
    - QTM performs all operations simultaneously and enters a superposition of all the resulting states
    - When QTM is measured, it collapses into a single complex number configuration (state) and behaves like the PTM upon observation



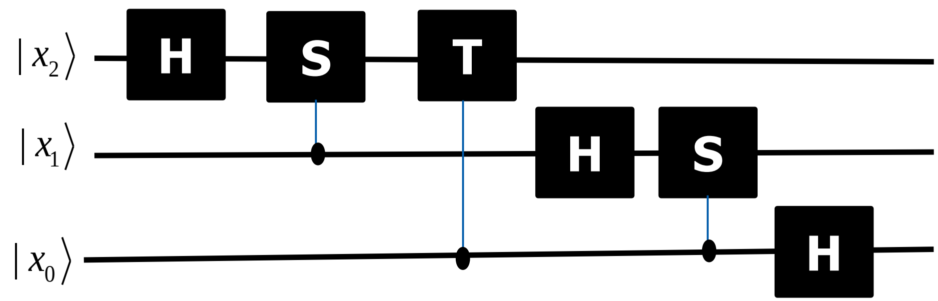
In "m" time steps the initial configuration will be in a configuration of "superposition(s) of configuration(s)":

$$\underbrace{U_M \circ U_M \circ ...U_M}_{t(m)\,times} |config_n\rangle = U_M^{t(m)} |config_n\rangle$$

Quantum Turing Complexity details in Backup Slides

Quantum Algorithms (continued)

- Quantum Fourier Transform (QFT) (Unitary Operator and Reversible)
  - n-qubit QFT
- Input State: $|\psi\rangle = \sum_{x=0}^{2n-1} \alpha_x |x\rangle$
- Output State: $|\psi'=\rangle = U_{QFT}|\psi\rangle = \sum_{x=0}^{2n-1}\sum_{y=0}^{2n-1} \frac{\alpha_x e^{2\pi ixy/2^n}}{\sqrt{2^n}} |y\rangle$



$$\frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2}\right)}|1\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^2}+\frac{x_1}{2}\right)}|1\rangle \otimes = \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^3}+\frac{x_1}{2^2}+\frac{x_2}{2}\right)}|1\rangle$$

- 3-qubit QFT
- Apply H gate to state $|x_2\rangle$

$$H|x_2\rangle = \frac{1}{\sqrt{2}}\sum_y (-1)^{x_2 y}|y\rangle = \frac{1}{\sqrt{2}}\sum_y e^{2\pi i x_2 y/2}|y\rangle$$

$$= \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^2}+\frac{x_2}{2}\right)}|1\rangle$$

- Apply S gate with control bit for state $|x_1\rangle$ either $|0\rangle$ or $|1\rangle$; $For\ |1\rangle:\ S|1\rangle = e^{2\pi i\frac{x_1}{4}}|1\rangle$
- State of System at this point:

$$I \otimes S|x_1\rangle = |x_1\rangle \otimes = \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^3}+\frac{x_1}{2^2}+\frac{x_2}{2}\right)}|1\rangle$$

- Apply T gate with control bit for state $|x_0\rangle$: $|x_0\rangle \otimes |x_1\rangle \otimes \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^3}+\frac{x_1}{2^2}+\frac{x_2}{2}\right)}|1\rangle$

- $|x_1\rangle$ goes through the H gate and Controlled S-gate: $|x_1\rangle \rightarrow \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^2}+\frac{x_1}{2}\right)}|1\rangle$

- State of System at this point:

$$|x_0\rangle \otimes = \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^2}+\frac{x_1}{2}\right)}|1\rangle \otimes = \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2^3}+\frac{x_1}{2^2}+\frac{x_2}{2}\right)}|1\rangle$$

- Finally Hadamard gate applied to $|x_0\rangle$: $|x_0\rangle \rightarrow \frac{1}{\sqrt{2}}|0\rangle + e^{2\pi i\left(\frac{x_0}{2}\right)}|1\rangle$

Final System State

Quantum Computing Algorithms (continued)

- Basic framework for all QC algorithms
  - Start with qubits in a particular classical state
  - The system is put into a superposition of many states
  - Unitary operations act on this superposition
  - Measurement of qubits in final states
- Definitions
  - **Discrete Logarithm Problem**: Given a prime number p, a base $b \in Z_p^*$, and an arbitrary element $y \in Z_p^*$, find an $x \in Z_p^*$ such that $b^x = y \bmod p$
  - **Hidden Subgroup Problem**: *G* is a group. Let *H* < *G* be a subgroup implicitly defined by a function of *f* on *G* is constant and distinct on every co-set o *H*. The problem is to find a set of generators for *H*
  - **Abelian Group (abstract algebra)**: Is a commutative group (generalize arithmetic addition of integers), is a group in which the result of applying the group operation to two group elements does not depend on the order in which they are written, i.e., these are the groups that obey the axiom of commutativity; named after early 19th century mathematician Niels Henrik Abel (ref. 21)
  - **Abelian Hidden Subgroup Problem**: *G* is a finite Abelian group with cyclic decomposition $G = Z_{n_0} \times \ldots \times Z_{n_L}$ Let *H* < *G* be a subgroup implicitly defined by a function of *f* on *G* is constant and distinct on every co-set o *H*. The problem is to find a set of generators for *H*
  - **Pell's Equation Problem**: Find an integral and positive solutions to $x^2 - dy^2 = 1$

Quantum Algorithms (continued)

- Grover's search algorithm (class of algorithms called *amplitude amplification*)
  - Finds an element in an unordered set quadratically faster $O(N^{1/2})$ time than any theoretical limit for classical algorithms O(N/2)
  - Internal calls to an oracle "O" for value of function (i.e., membership is true for an instance)

N entries with *n = log(N)* bits

Apply Hadamard transform on $|0\rangle^{\otimes n}$ to produce equal superposition state

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{x=0}^{n-1} |x\rangle$$

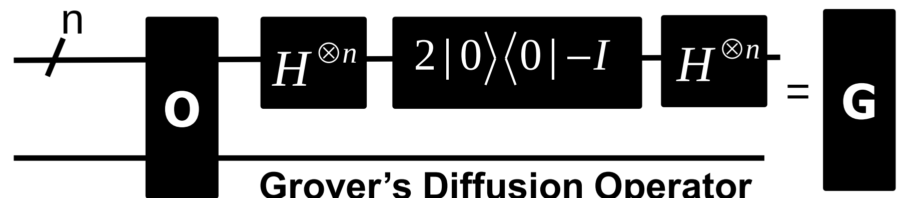
Apply the Grover diffusion operator

2 Hadamard operations require *n* operations each

The conditional phase shift is a controlled unitary operation and require *O(n)* gates

The Oracle complexity is application dependent, in this algorithm it requires only one call per iteration

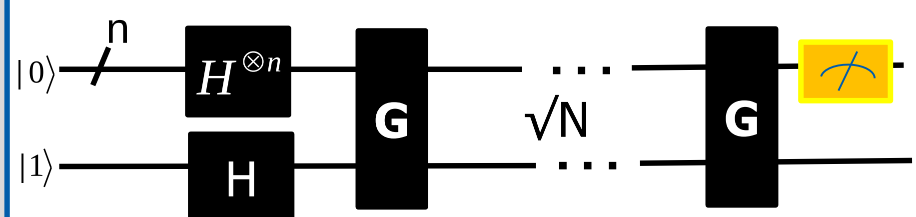
Apply measurement



Grover's Diffusion Operator

1 Apply a call to Oracle O
2 Apply the Hadamard transform $H^{\otimes n}$
3 Apply a phase shift (excluding $|0\rangle$): $|x\rangle \rightarrow -(-)^{\phi_{x0}} |x\rangle$
4 Apply Hadamard transform $H^{\otimes n}$

$$H^{\otimes n}\left(2|0\rangle\langle 0| - I\right)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$$



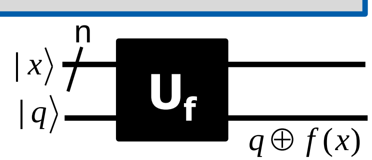
Grover's Circuit / Algorithm

Start: Initialize the n-qubit states to $|0\rangle^{\otimes n}$

Identify: Element requested (ensure it is available)
- Apply Hadamard transform to n-quits and initialize superposition $H^{\otimes n}|0\rangle^{\otimes n}$
- for O(√N) times do
  - Apply the Grover operator G
- end for

Measure the system

Oracle



Quantum Algorithms (continued)

- Quantum Fourier Transform (QFT) (Unitary Operator and Reversible)

n-qubit QFT

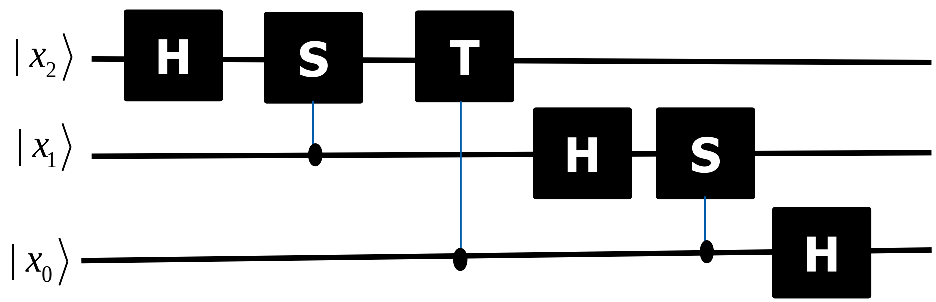
- Input State: $|\psi\rangle = \sum_{x=0}^{2n-1} \alpha_x |x\rangle$
- Output State: $|\psi' =\rangle = U_{QFT}|\psi\rangle = \sum_{x=0}^{2n-1}\sum_{y=0}^{2n-1} \frac{\alpha_x e^{2\pi ixy/2^n}}{\sqrt{2^n}} |y\rangle$

$$O(\log^2 n)\ execution\ time$$

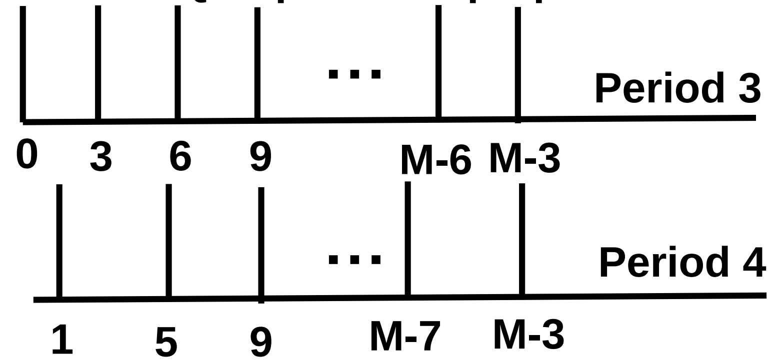
$$\omega = e^{\frac{\pi i}{4}}$$

$\mathbf{U_{QFT}}$

$$\frac{1}{\sqrt{2^n}}\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{2^n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(2^n-1)} \\ 1 & \omega^3 & \omega^6 & \cdots & \omega^{3(2^n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2^n-1} & \omega^{2(2^n-1)} & \cdots & \omega^{(2^n-1)(2^n-1)} \end{pmatrix}$$



QFT period superpositions



2-Qubit & 3-Qubit treatment in Backup Slides

Quantum Algorithms (continued)

- 2 Qubit QFT matrix form
  - QFT full matrix form:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle;$$

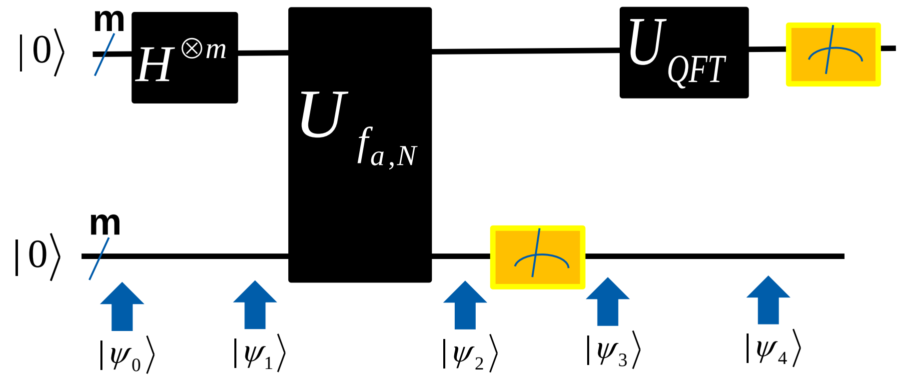
$$|\psi'\rangle = U_{QFT}|\psi\rangle = \frac{1}{\sqrt{4}}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{\frac{\pi i}{4}} & e^{\frac{\pi i2}{4}} & e^{\frac{\pi i3}{4}} \\ 1 & e^{\frac{\pi i2}{4}} & e^{\frac{\pi i4}{4}} & e^{\frac{\pi i6}{4}} \\ 1 & e^{\frac{\pi i3}{4}} & e^{\frac{\pi i6}{4}} & e^{\frac{\pi i9}{4}} \end{pmatrix}\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$= \frac{1}{\sqrt{4}}\begin{pmatrix} \frac{1}{\sqrt{2}}+0+0+\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{1}{\sqrt{2}}e^{\frac{\pi i3}{4}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{1}{\sqrt{2}}e^{\frac{\pi i6}{4}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{1}{\sqrt{2}}e^{\frac{\pi i9}{4}} \end{pmatrix} = \frac{1}{\sqrt{4}}\begin{pmatrix} \frac{1}{\sqrt{2}}+0+0+\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{-1+i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{-i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}+0+0+\frac{1+i}{\sqrt{2}} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{2}{\sqrt{8}} \\ \frac{\sqrt{2}-1+i}{4} \\ \frac{1-i}{\sqrt{8}} \\ \frac{\sqrt{2}+1+i}{4} \end{pmatrix} = \frac{2}{\sqrt{8}}|00\rangle + \frac{\sqrt{2}-1+i}{4}|01\rangle + \frac{1-i}{\sqrt{8}}|10\rangle + \frac{\sqrt{2}+1+i}{4}|11\rangle.$$

Quantum Algorithms (continued)

- Shor's algorithm
  - Is a factoring algorithm
    - It can be used to break encryption codes
  - Computation execution time is $O(n^2 \log n \log \log n)$ number of polynomial steps; n bits to represent number N
  - Classically it is $O(e^{cn^{1/3} \log^{2/3} n})$ exponential steps



$$|\psi_0\rangle = |0_m, 0_n\rangle; \quad |\psi_1\rangle = \frac{\sum_{x\in\{0,1\}^m} |x, 0_n\rangle}{\sqrt{2^m}};$$

Evaluation of f *on all possibilities:*

$$|\psi_2\rangle = \frac{\sum_{x\in\{0,1\}^m} |x, f_{a,N}(x)\rangle}{\sqrt{2^m}} = \frac{\sum_{x\in\{0,1\}^m} \left|x, a^x Mod(N)\right|}{\sqrt{2^m}};$$

$$|\psi_3\rangle = \frac{\sum_{a\equiv a^{\bar{x}} Mod(N)} \left|x, a^{\bar{x}} Mod(N)\right|}{\left[\frac{2^m}{r}\right]} = \frac{\sum_{j=0}^{\frac{2^m}{r}-1} \left|t_0 + jr, a^{\bar{x}} Mod(N)\right|}{\left[\frac{2^m}{r}\right]};$$

where t_0 *is the first time* $a^{t_0} = a^{\bar{x}} Mod(N)$ *is measured*

Algorithm Steps

1. Input a positive integer N with $n=\log_2 N$
2. Use a polynomial algorithm to determine if N is a prime or a power of prime. If it is prime, declare and exit. If it is power of prime, declare and exit
3. Randomly select an integer a: 1<a<N. Perform Euclid's algorithm to find GCD(a,N). If GCD is not 1, then return value and exit
4. Use the quantum circuit to find the period r
5. If r is odd, or if $a^{r/2} \equiv -1$ Mod N return to Step 3 and choose another a
6. Use Euclid's algorithm to calculate the GCD($a^{r/2}$
7. + 1,N) and GCD($a^{r/2}$ - 1,N). Return at least one non trivial solution
8. Output a factor p of N if it exists

Quantum Adiabatic Computing

- Uses adiabatic processes for QC in the following steps:
  - Create an initial state of qubits
  - Start with an initial Hamiltonian and very it very slowly (adiabatically)
    - $H_{initial}$ transforms into H_{final} whose eigenstates encode the solution
  - The Hamiltonian ground state is created

$$H_{initial} = -\sum_j X^j$$

    - Consists of Pauli Operators
  - The final Hamiltonian

$$H_{final} = -\sum_x c_x \,|\, x \rangle \langle x \,|$$

  - If the T is the total time of computation, we can interpolate the Hamiltonian solution at any time "t". Let s=1/T with 0≤s≤1:

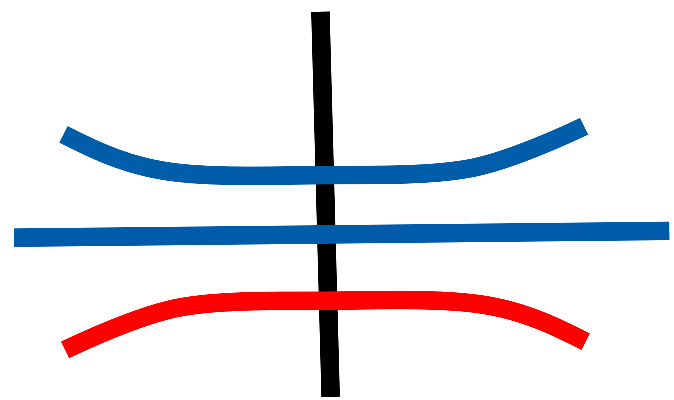
$$\hat{H} = (1-s)H_{initial} + sH_{final}$$

Adiabatic Process

$$t_{Hamiltonian} \gg t_{Critical};\ H_{initial} \underset{slowly}{\rightarrow} H_{final}$$

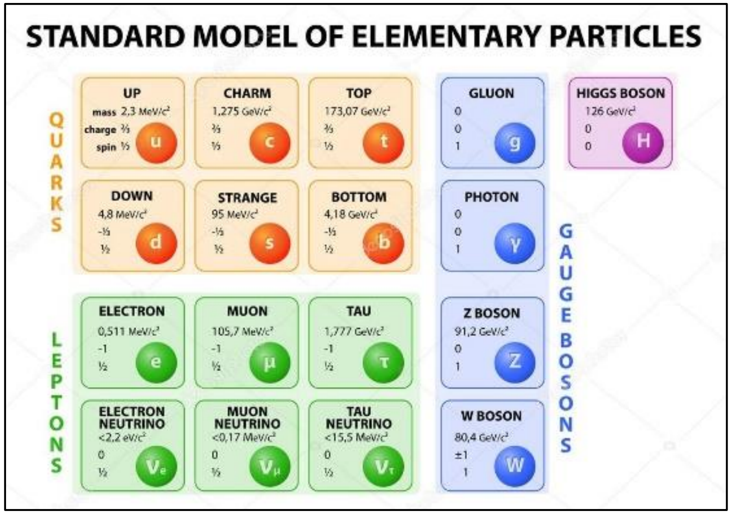
$$Uncertainity\ \Pr inciple: \Delta E \Delta t \geq \frac{\hbar}{2}$$

$$\Rightarrow \Delta t \geq \frac{\hbar}{2}\frac{1}{\Delta E}$$



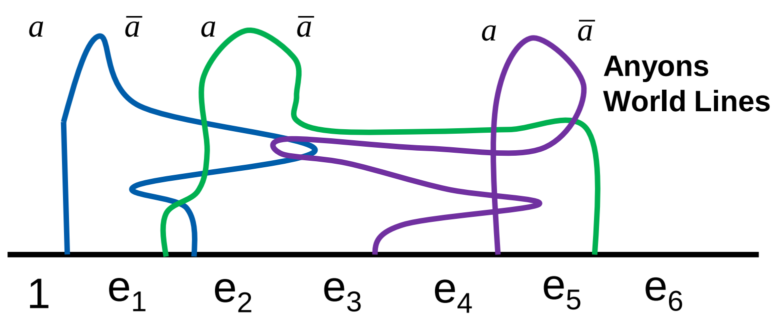
Topological Quantum Computing (QC)

- Anyons (named by Frank Wilczek 1982 – ref. 19)



  - Obey exotic statistics including Fermi-Dirac statistics for fermions (Leptons, Quarks)
  - Bose-Einstein statistics for bosons (Gauge, Higgs)
  - They cannot occupy the same space
  - Have arbitrary phase factors
  - Follow non-trivial unitary evolutions when particles are exchanged
  - Transformation of the anionic wave function obey exchange symmetry
  - Hence the name “Any” + “ons”
- Kitaev (2003 – ref. 20) demonstrate that anyons could be used to perform fault tolerant computation

Anyonic QC	
QC	**Anyonic Operations**
Initialize state	Create and arrange anyons
QC gates	Braid anyons
State measurement	Detect anionic charge



- One configuration of topological fault tolerant quantum computation
- During initialization a pair of anyons $a, \bar{a}$ are created from vacuum (i.e., e^-, e^+ electron-positron pair)
- Braided operations unitarily evolve to their fusion state
- Fusing the anyons together give a set of measurement outcomes e_i; i=1,… which encodes the results of the computation

Laboratory Systems: Electron gas in high magnetic field is sandwiched between thin semiconductor layers of aluminum gallium arsenide

Cluster State Quantum Computing (CSQC)
Represent CSQC as Graphs

NORTHROP GRUMMAN

- CSQC is a multipartite qubit (highly entangled) modeling scheme. It simulates unitary dynamics in crystal lattices. Within this model, the cluster states are a series of measured points in the computation; the result is used to select a new basis for the next measurement, thus forming a *feedback loop*
  - CSQC is represented a graph (each node/vertex of the graph is a qubit; the edges of the graph are the CZ gates
  - It is a two-step process: 1) initialize a set of qubits in some state, for example start with |+> then apply the CPHASE gates to the states
  - Measure the qubits in some basis states. As the next measurement is taken the choice of the new basis depends/determined by the previous measurement results
  - Effect of CZ application:

$$\text{Example: intial } \textit{product} \text{ state: } |+\rangle_C = |+\rangle \otimes |+\rangle \to CZ\,|+\rangle \otimes |+\rangle,$$

$$\text{where } \; CZ = \frac{1}{2}[I \otimes I + Z \otimes I + I \otimes Z - Z \otimes Z]$$

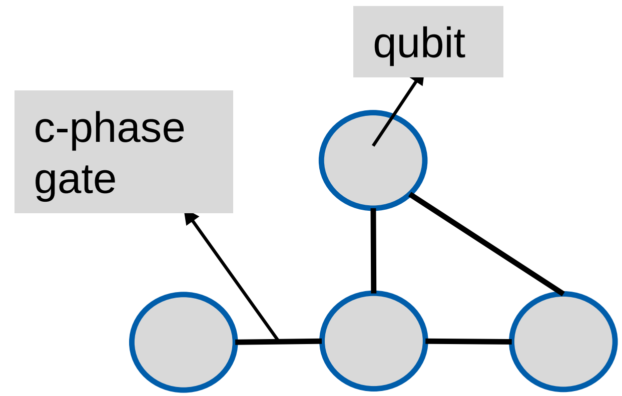
Chose a basis state:

$$A = \frac{1}{2}\left[\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\right]$$

$$CZ\,|+\rangle \otimes |+\rangle = [A + (Z \otimes I)A + (I \otimes Z)A - (Z \otimes Z)A]$$

$$= \frac{1}{2}[|00\rangle + |01\rangle + |10\rangle - |11\rangle]$$

  - This operation gives an entangled 2-Qubit State represented by: 



4-qubit cluster state
Edges are c-phase gates
Vertices are qubits

$$CZ \; (controlled \;\; Z \;\; gate$$
$$is \; controlled \;\; phase \; operation):$$

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

Phase shift is applied to the target qubit with control qubit in state |1>: CZ|11>=-|11>

Quantum Particle Swarm Optimization (QPSO)

From: Jun Sun, Choi-Hong Lai, Xiao-Jun Wu, "Particle Swarm Optimisation-Classical and Quantum Perspective", Chapman & Hall/CRC Press, 2012

- QPSO Algorithm
  - Uses the one of many potential functions for determination of particle position using the Schrödinger equation with Hamiltonian Ĥ (here the simple case of delta potential well is used)
  - Uses the mean best position "x" of particle to enhance the global search capability for particle position
  - Unlike the classical PSO algorithm the QPSO does not require the velocity vectors of particle and fewer parameters to adjust. It is simpler to implement
  - Choosing QPSO parameters swarm size, problem dimension, the number of maximum iteration, and the most important parameter "α" the contraction-expansion coefficient (CE) describes the dynamical behavior of individual particles and the algorithm converges (for $\alpha \le \alpha_0 \in [1.7, 1.8]$)
  - i.e., $\alpha_0 = e^{\gamma} = 1.781$; is optimized for behavior particle $\Rightarrow \gamma = 0.577215665$ is called the Euler constant

- qpso\qpso.bat finds the mean best fit to particle position "x"

$$\hat{H} = -\frac{\hbar}{2m}\frac{d}{dy^2} - \gamma\delta(y);$$

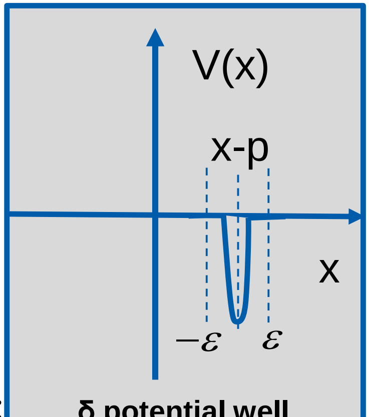
where γ is intensity of potential well, $y = x - p$;

Schrödinger equation:

$$\frac{d^2\psi}{dy^2} + \frac{2m}{\hbar^2}\left[E + \gamma\delta(y)\right]\psi = 0;$$

Wave function solution is:

$$\psi(y) = \frac{1}{\sqrt{L}} e^{-\frac{y}{L}};\ L = \frac{1}{\beta} = \frac{\hbar^2}{m\gamma};$$



δ potential well

Probability Distribution Function:

$$F(y) = 1 - e^{-2\frac{|y|}{L}};$$

Particles position is given by:

$$x = p \pm \frac{L}{2}\ln\left(\frac{1}{u}\right);$$

where u is random number uniformly distributed on (0,1); $u \sim U(0,1)$

QPSO (continued)

- Variants of QPSO have been utilized
  - Cooperative QPSO (CQPSO); Gao et. Al [2007] , Sun et al. [2008]
  - Diversity-controlled QPSO (DCQPSO); Riget et al. [2002], Ursem et al. [2001], Sun et al. [2006]
  - Local-attractor QPSO (LAQPSO); Shao et al. [2016]
  - QPSO Tournament-selector (QPSO-TS); P. Angeline [1998]
  - QPSO-Roulette-Wheel selection (QPSO-RS); Long et al. [2009]
  - QPSO with Hybrid Distribution (QPSO-HD); Sun et al. [2006]
  - QPSO with Mutation; Liu et al. [2006], Fang et al. [2009]

- H. Gao et al., A cooperative approach to quantum-behaved particle swarm optimization, In Proceedings of the 2007 IEEE International Symposium on Intelligent Signal Processing, Madrid, Spain, 2007, pp. 1–6
- S. Lu, C. Sun, Quantum-behaved particle swarm optimization with cooperative-competitive coevolutionary, In Proceedings of the 2008 International Symposium on Knowledge Acquisition and Modeling, Wuhan, China, 2008, pp. 593–597. 32
- S. Lu, C. Sun, Coevolutionary quantum-behaved particle swarm optimization with hybrid cooperative search, In Proceedings of the 2008 Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Washington, DC, 2008, pp. 109–113
- J. Sun, W. Xu, W. Fang. Quantum-behaved particle swarm optimization with a hybrid probability distribution, In Proceedings of the Ninth Pacific Rim International Conference on Artificial Intelligence, Guilin, China, 2006, pp. 737–746.
- Shao D., Hu S., Fei Y., A new quantum particle swarm optimization algorithm, Neural Network World 5/2016, 477–496
- J. Liu, J. Sun, W. Xu. Quantum-behaved particle swarm optimization with adaptive mutation operator, In Proceedings of the 2006 International Conference on Natural Computing, Hainan, China, 2006, pp. 959–967.
- W. Fang, J. Sun, W. Xu. Analysis of mutation operators on quantum-behaved particle swarm optimization algorithm, New Mathematics and Natural Computation, 2009, 5(2): 487–496
- H. Long, J. Sun, X. Wang, C. Lai, W. Xu. Using selection to improve quantum behaved particle swarm optimization, International Journal of Innovative Computing and Applications, 2009, 2(2): 100–114.
- P.J. Angeline, Using selection to improve particle swarm optimization, In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, Anchorage, AK, 1998, pp. 84–89.
- J. Riget, J. Vesterstroem. A diversity-guided particle swarm optimizer—The ARPSO: Department of Computer Science, University of Aarhus, Aarhus, Denment, 2002
- R.K. Ursem. Diversity-guided evolutionary algorithms, In Proceedings of the 2011 Parallel Problem Solving from Nature Conference, Paris, France, 2001, pp. 462–471
- J. Sun, W. Xu, W. Fang, Quantum-behaved particle swarm optimization algorithm with controlled diversity, In Proceedings of the 2006 International Conference on Computational Science, Reading, MA, 2006, pp. 847–854.
- J. Sun, W. Xu, W. Fang, Enhancing global search ability of quantum-behaved particle swarm optimization by maintaining diversity of the swarm, In Proceedings of the 2006 International Conference on Rough Sets and Current Trends in Computing, Kobe, Japan, 2006, pp. 736–745.

QPSO (continued)
Applications

- **Antenna Design:** Determine infinitesimal dipoles to represent an arbitrary antenna for near-field distributions (ref. Mikki et al. [2006])
- **Biomedicine:** Coupling RFB neural networks to the QPSO algorithm for the culture conditions of hyaluronic acid production by Streptococcus zooepidemicus (Lui et al. [2009]). Lu and Wang [2008] employed QPSO to estimate parameters from kinetic model of batch fermentation
- **Mathematical Programming:** Integer programming (Liu et al. [2006]), constrained non-linear programming (Liu et al. [2008]), combinatorial optimization (Wang et al. [2008]), layout optimization (Xiao et al. [2009]), and multiobjective design optimization of laminated composite components (Omkar et al. [2009])
- **Communication Networks:** NP-hard QoS multicast routing (converted to integer programming and solved by Sun et al. [2006]), RBFNN network anomaly detection (hybrid QPSO with gradient descent algorithm to train RBFNN by Ma et al. [2008], Wavelet NN & conjugate gradient algorithm for network anomaly detection (Ma et al. [2007], WLS-SVM QPSO for anomaly detection (Wu et al. [2008]), mobile IP routing (Zhao et al. [2008]), and channel assignment (Yue et al [2009])

- S. Mikki et al., Infinitesimal dipole model for dielectric resonator antennas using the QPSO algorithm, Proceedings of the 2006 IEEE Antennas and Propagation Society International Symposium, Albuquerque, NM, 2006, pp. 3285-3288
- Lui et al., Culture conditions…neural network and quantum-behaved particle swarm optimization algorithm, Enzyme and Microbial Technology, 2009, 44(1), pp. 24-32
- K. Lu and R. Wang, Application of PSO and QPSO … glutamic acid batch fermentation, In Proceedings of the Seventh World Congress on Intelligent Control and Automation, Chongqinq, China, 2008, pp. 8968-8971
- J. Liu et al., Quantum-behaved particle swarm optimization for integer programming, In Proceedings of the 2006 International Conference on Neural Information Processing, Hong Kong, China, 2006, pp. 1042-1050
- H. Liu et al., A modified quantum-behaved particle swarm optimization for constrained optimization, In Proceedings o the 2008 International Symposium on Intelligent Information Technology Application Workshops, Shanghai, China, 2008, pp. 531-534
- J. Wang et al., Discrete quantum-behaved particle swarm optimization of distribution for combinatorial optimization, In Proceedings of the 2008 IEEE World Congress on Computational Intelligence, Hong Kong, China, 2008, pp. 897-904
- B. Xiao et al., Optimal planning of substation locating and sizing based on improved QPSO algorithm, In Proceedings of the Asia-Pacific, Power and Energy Engineering Conference, Shanghai, China, 2009, pp. 1-5

QPSO (continued)
Applications

- Many **other** applications employing QPSO algorithm in the following areas:
  - **Control Engineering**
  - **Clustering & Classification**
  - **Image Processing**
    - **Image processing, image segmentation, image registration, image interpolation, and face recognition and registration**
  - **Fuzzy Systems**
  - **Finance**
  - **Graphics**
    - **Rectangular packing problem, polygonal approximation curves, and irregular polygon layouts**
  - **Power Systems**
  - **Modelling**
    - **SVM, LS-SVM**
    - **Transistor Devices**
    - **Detection of unstable orbits in a non-Lyapunov technique**
  - **Filters**
    - **Design of Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters**
  - **Multiprocessor Scheduling**

- S.N. Omkar et al., Quantum behaved particle swarm optimization (QPSO) for multi-objective design optimization of composite structures, Expert Systems with Applications, 2009, 36(8), pp. 11312-11322
- R. Ma et al., Network anomaly detection using RBF neural networks with hybrid QPSO, In Proceedings of the IEEE International Conference on Networking, Sensing and Control, Chicago, IL, 2008, pp, 1284-1287
- J. Sun et al., QoS multicast routing algorithm, In Proceedings of the 2006 International Conference on Simulated Evolution and Learning, Hefei, China, 2006, pp. 261-268
- D. Zhao et al., An approach to mobile IP routing based on QPSO algorithm. In Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China, 2008, pp. 667-671
- R. Ma et al., Hybrid QPSO based wavelet neural networks for network anomaly detection, In Proceedings of the Second Workshop on Digital Media and its Application in Museum and Heritages, Qingdao, China, 2007, pp 442-447
- R. Wu et al., An approach to WLS-SVM based on QPSO algorithm in anomaly detection, In Proceedings of the 2008 World Congress on Intelligent Control and Automation, Chongqing, China, 2008, pp. 4468-4472
- C. Yue et al., Channel assignment based on QPSO algorithm, Communications Technology, 2009, 42(2), pp. 204-206

From: Jun Sun, Choi-Hong Lai, Xiao-Jun Wu, "Particle Swarm Optimisation-Classical and Quantum Perspective", Chapman & Hall/CRC Press, 2012

THE VALUE OF PERFORMANCE.