Introduction to Quantum Computing

Jian Tao

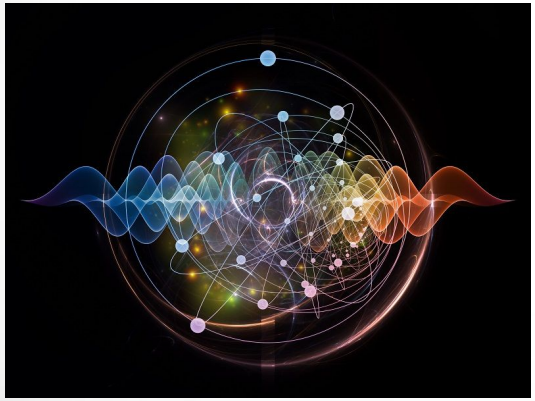
jtao@tamu.edu

Spring 2019 HPRC Short Course

04/05/2019

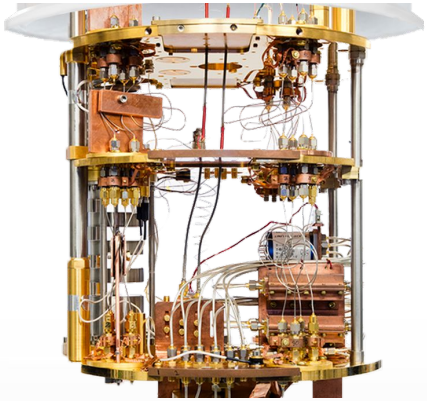
Outline

SCIENTIFIC PRINCIPLES



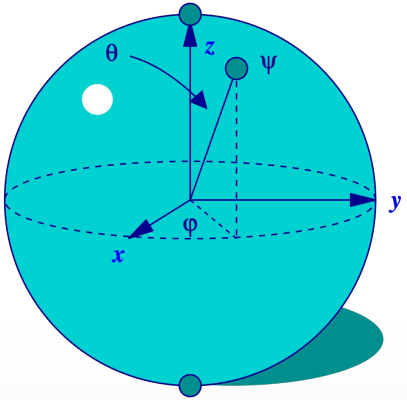
Quantum Physics

HARDWARE



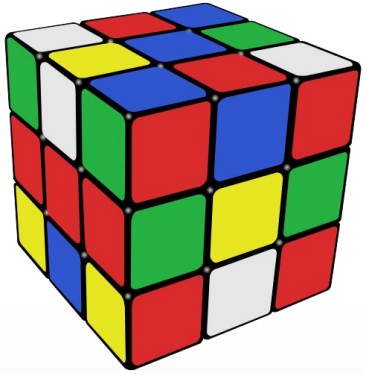
Quantum Computer

MATHEMATICAL THEORY



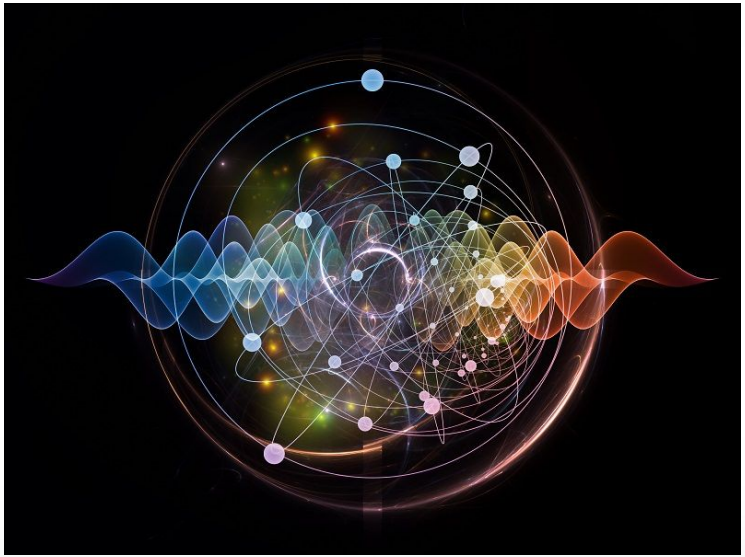
Quantum Information

SOFTWARE IMPLEMENTATION



Quantum Programming

Quantum Physics



(Image: Queen's University Belfast)

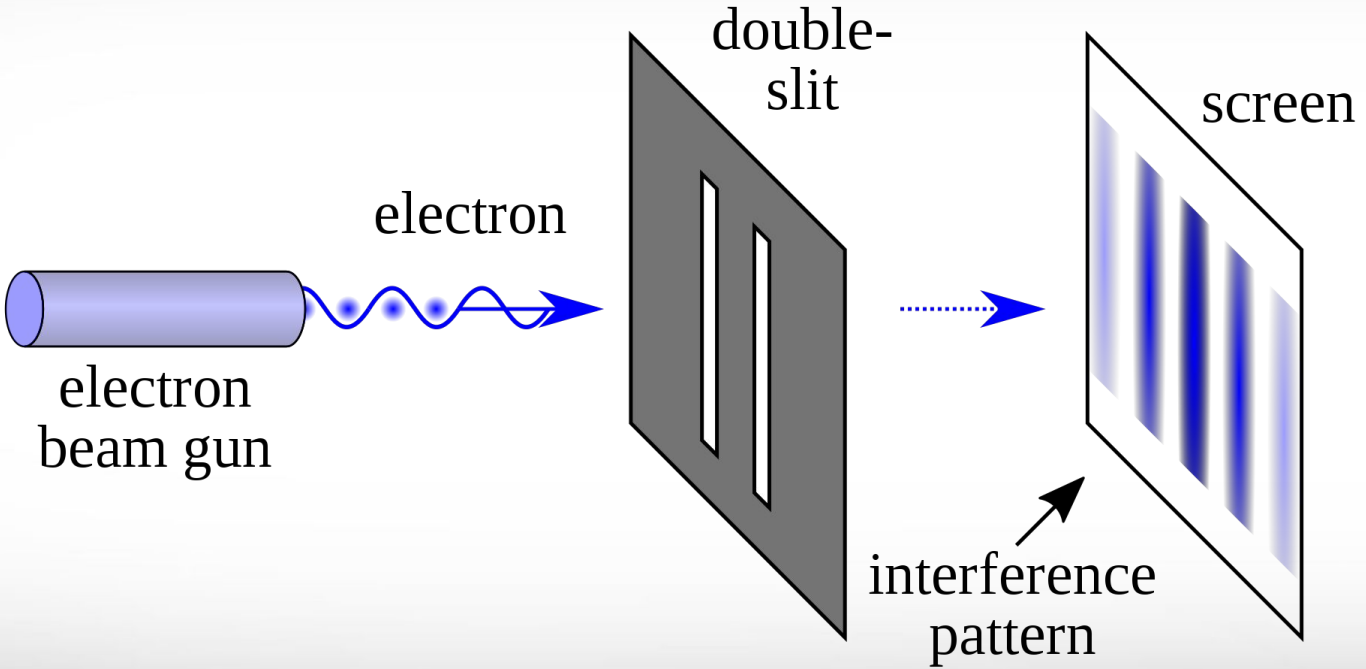
Quantum Physics

"I think I can safely say that nobody understands quantum mechanics."

- Richard Feynman

Richard Feynman Messenger Lectures - Lecture 6 Probability and Uncertainty: The Quantum Mechanical View of Nature | The Character of Physical Law | Richard Feynman (1964)
http://www.cornell.edu/video/playlist/richard-feynman-messenger-lectures

Double Slit Experiment - Particle Duality



(Image: Wikipedia)

Heisenberg's Uncertainty Principle

$$\Delta x \Delta p \geq \frac{\hbar}{2}$$

The principle asserts a fundamental limit to the precision with which certain pairs of physical properties of a particle, known as complementary variables or canonically conjugate variables such as position x and momentum p, energy E and time t, can be known.

Schrödinger's Equation

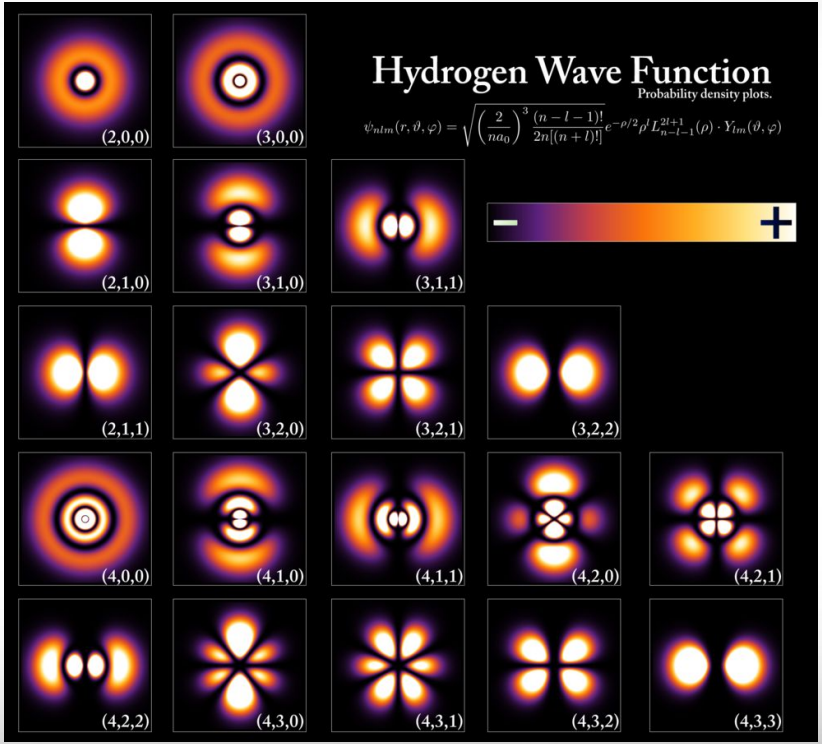
$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

Schrödinger's equation is **THE EQUATION** that all applications of quantum mechanics follow. Quantum physics is primarily about studying, solving, and applying Schrödinger's equation and its extensions under certain physical conditions.

Hydrogen Wave Function

The hydrogen wave function depends on three quantum numbers **n**, **m**, and **l**. $Y_{\ell m}(\theta, \phi)$ is the spherical harmonics that depends only on the angular coordinates.

$$\psi_{n\ell m}(r,\theta,\phi) = \sqrt{\left(\frac{2}{na_0}\right)^3 \frac{(n-\ell-1)!}{2n[(n+\ell)!]}} e^{-r/na_0} \left(\frac{2r}{na_0}\right)^{\ell} L_{n-\ell-1}^{2\ell+1}\left(\frac{2r}{na_0}\right) \cdot Y_{\ell}^{m}(\theta,\phi)$$



(Image: Wikipedia)

Quantum Superposition

Since Schrödinger's equation is **LINEAR**, any linear combination of solutions will also be a solution. Given two "basis states"

$$|0\rangle \qquad |1\rangle$$

a superposition of the basis states can be given by

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ is the probability of outcome |0> and $|\beta|^2$ is the probability of outcome |1>

Quantum Entanglement

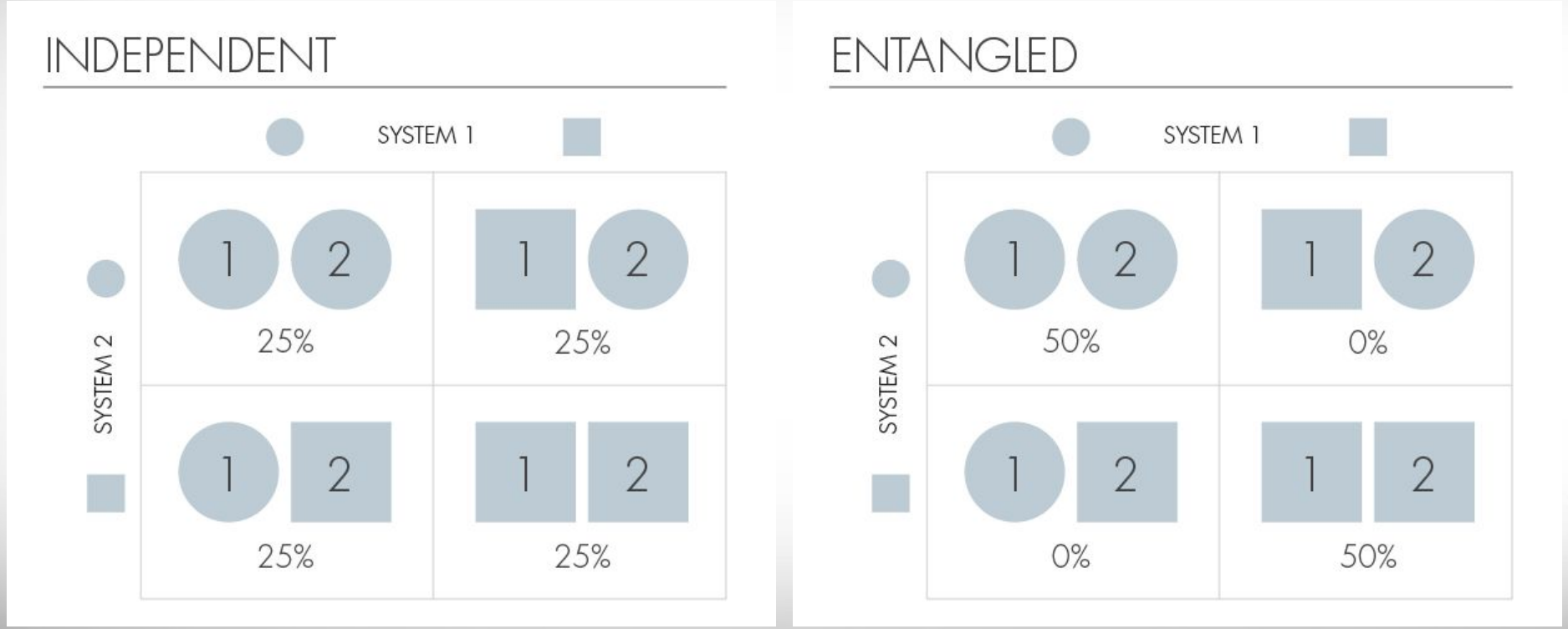
When groups of particles are generated, they interact, or share spatial proximity so that the quantum state of each particle cannot be described independently of the state of the other(s). For two qubits,

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

For example, two entangled qubits is

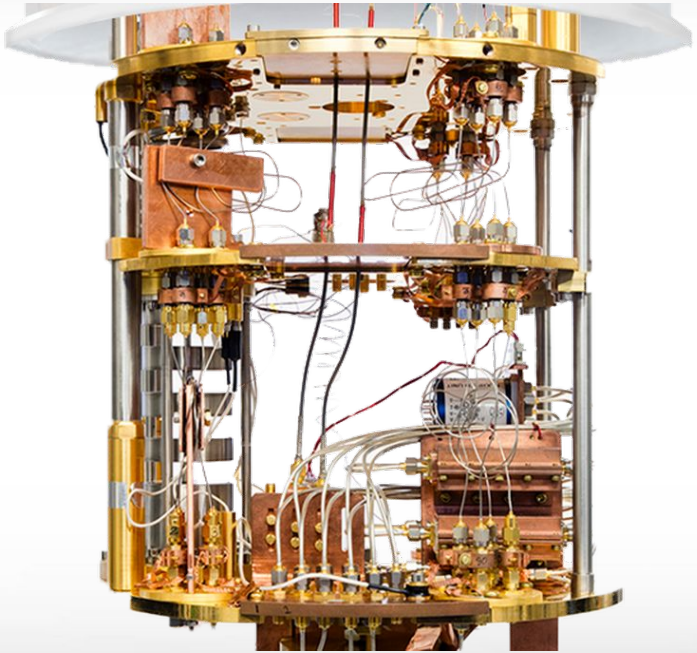
$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Quantum Entanglement



(Image: Olena Shmahalo/Quanta Magazine)

Quantum Computer



(Image: IBM Q)

Quantum Computing - Overview

"Quantum computing is the use of quantum-mechanical phenomena such as **superposition** and **entanglement** to perform computation." - Wikipedia

"Every finitely realizable physical system can be perfectly simulated by a **universal model computing machine** operating by **finite means**." - David Deutsch

Quantum Computers - Current Status

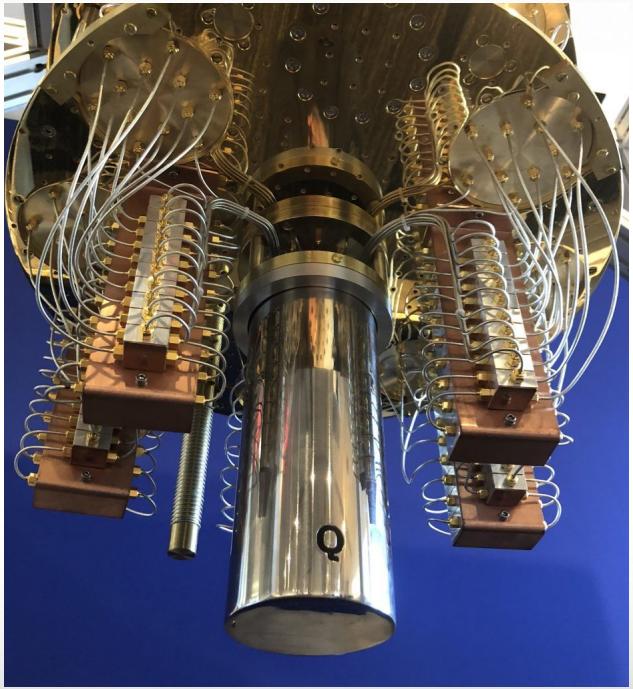
Quantum Processors (as of April 2019)

D-Wave 2000Q	Superconducting	2048 qb	2017
19Q Acorn	Superconducting	19 qb	17 December 2017
Tangle Lake	Superconducting	49 qb	9 January 2018
IBM Q 50 prototype	Superconducting	50 qb	N/A
Digital Annealer	Digital Circuit	8,192 bit	21 December 2018
Bristlecone	Superconducting	72 qb	5 March 2018
Station Q	superconducting	N/A	N/A

(Table: Wikipedia)

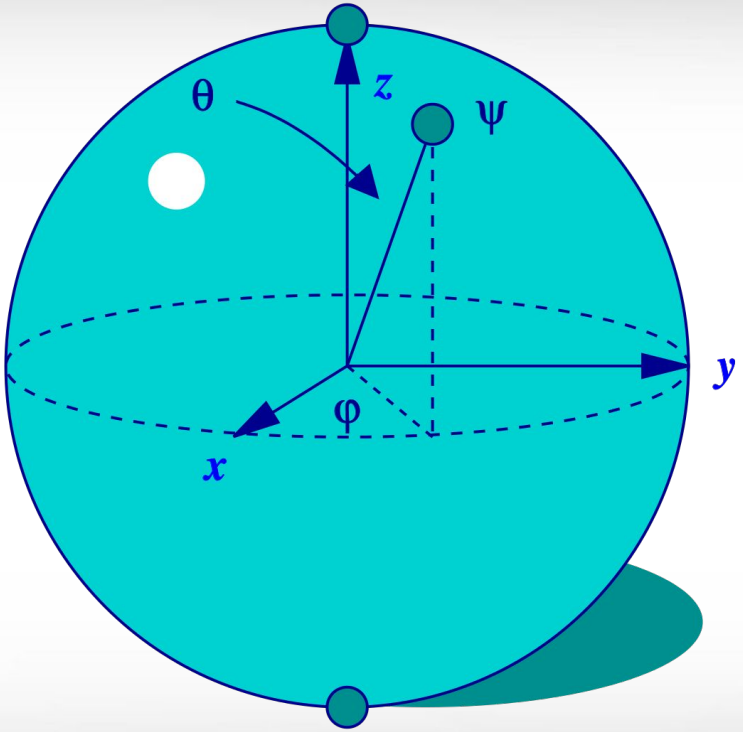
Quantum Processor - Hardware

- Very small (< 10^{-9}m, at the atomic scale)
- Isolated (not in close proximity with anything else to reduce noise)
- Very cold (~10mK, close to absolute zero)



(Image: IBM Q)

Quantum Information



(Image: Bloch Sphere - Wikipedia)

Qubits - Quantum Bits

A single qubit can be described by a linear combination of $|0\rangle$ and $|1\rangle$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Qubits States

1 qubit	2 states
2 qubits	4 states
3 qubits	8 states
4 qubits	16 states
10 qubits	1024 states

Physical Realization of Qubits

PHYSICAL REALIZATION	QUBIT IMPLEMENTED BY
Superconducting	the state of small superconducting circuits
Trapped ion	the internal state of trapped ions
Optical lattices	the internal state of neutral atoms trapped in an optical lattice
Quantum dot	the spin state of trapped electrons or electron position in double quantum dot.
Nuclear magnetic resonance	the nuclear spin within the dissolved molecule and probed with radio waves
Split electron with Majorana fermion	electron fractionalization and ground state degeneracy

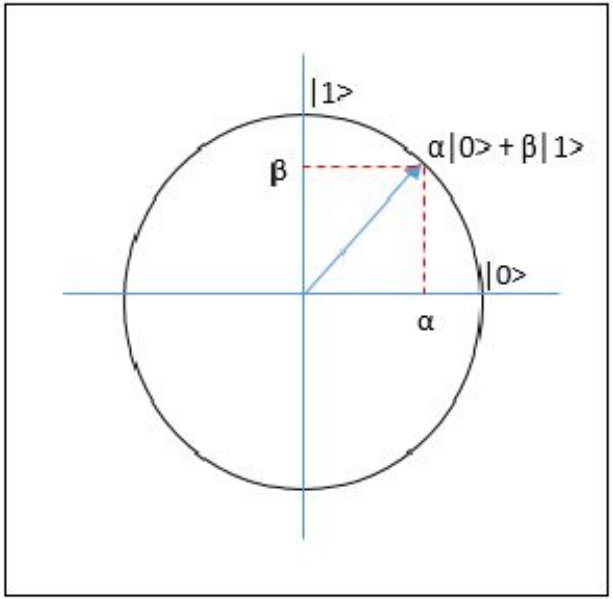
Qubits - Standard Representation

The general quantum state of a qubit can be represented by a linear superposition of its orthonormal basis states (or basis vectors). For a two qubit system, the vectors are usually denoted as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Qubits - Graphical Representation

States in which qubit are measured are known as basis vectors. Qubits are represented as |0> and |1>



(Image: IJSER)

Quantum Logic Gate

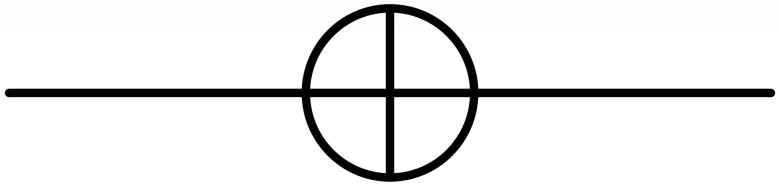
A unitary operator that acts on a small number of qubits is often called a **GATE**, in analogy to classical logic gates like AND, NOT, etc.

Hadamard (H) gate
Pauli-X or NOT gate
Pauli-Y gate
Pauli-Z gate
Square root of NOT gate
Phase shift gates
SWAP gate
Square root of Swap gate

Controlled cX or CNOT gate
Controlled (cY cZ) gates
Toffoli (CCNOT) gate
Fredkin (CSWAP) gate
Ising (XX) coupling gate
Ising (YY) coupling gate
Ising (ZZ) coupling gate
Deutsch gate

Not Gate or Pauli-X Gate

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

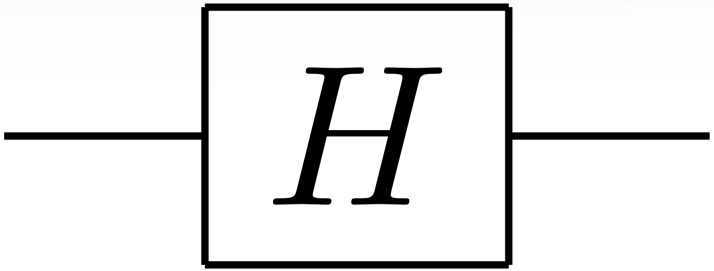


$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

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

Hadamard Gate - "Superposition" Gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

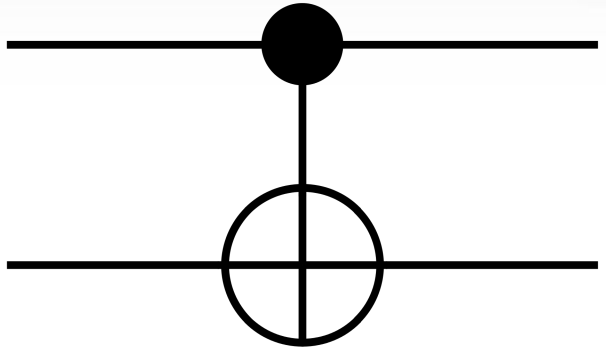


$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

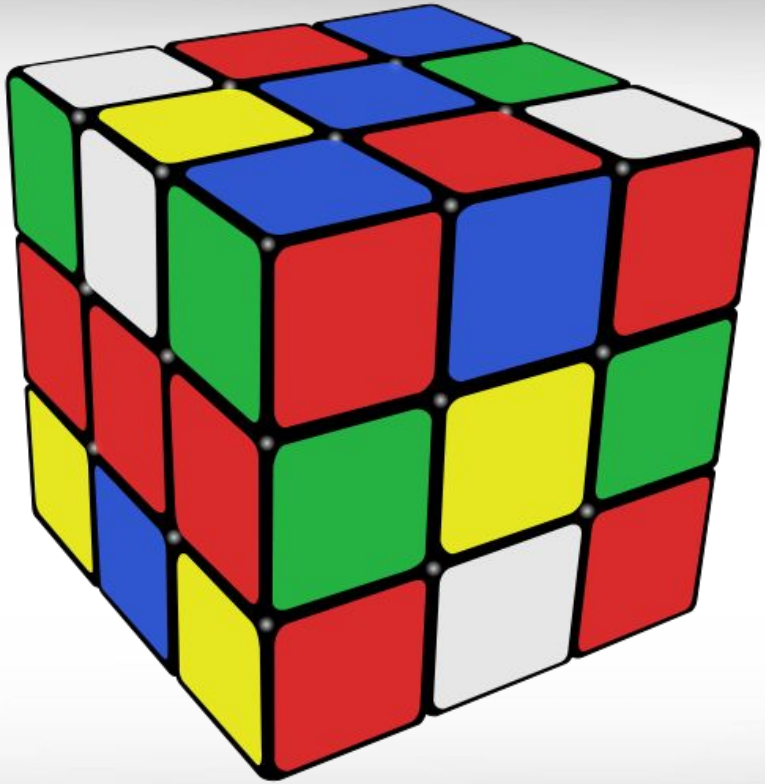
CNOT Gate - Controlled Not or cX Gate

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



The **CNOT** gate is generally used in quantum computing to generate entangled states.

Quantum Programming

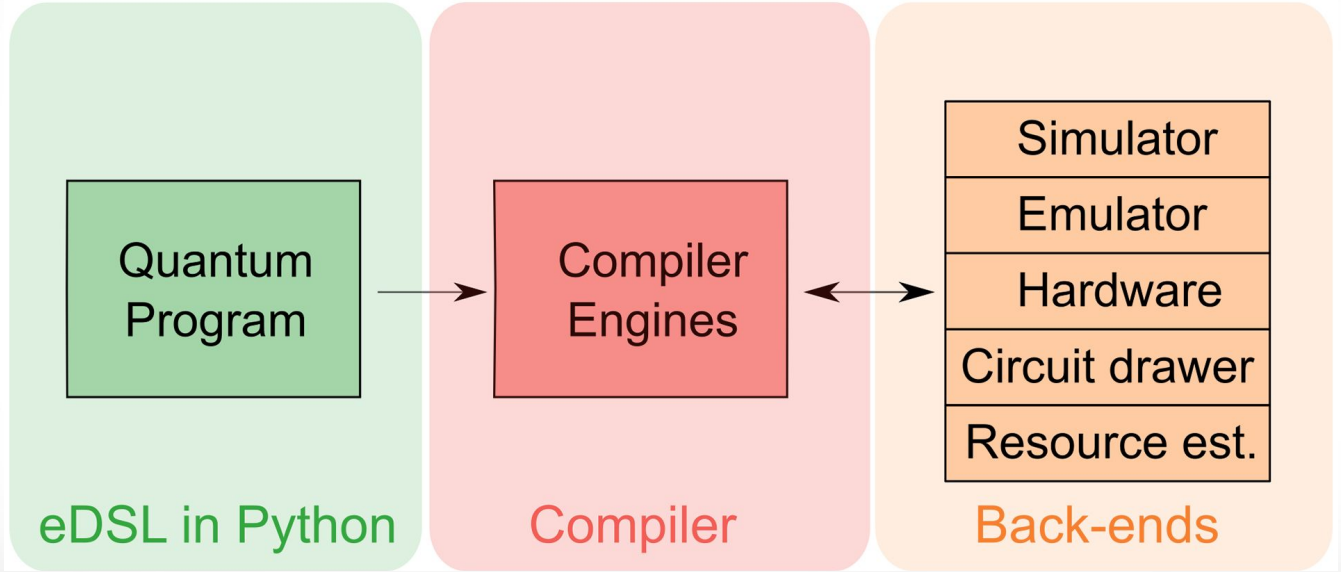


(Image: Wikipedia)

Full Stack Quantum Computing Libraries

Qiskit	IBM	Python	Framework for working with noisy quantum computers at the level of pulses, circuits, and algorithms
pyQuil	Rigetti	Python	Rigetti's software library for writing, simulating, compiling and executing quantum programs.
Ocean	D-Wave	Python	D-Wave System's suite of tools for solving hard problems with quantum computers.
ProjectQ	IBM	Python	Hardware-agnostic framework with compiler and simulator with emulation capabilities.
Cirq	Community Code	Python	Cirq is a Python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators.
Strawberry Fields	Xanadu	Python	Strawberry Fields is a full-stack Python library for designing, simulating, and optimizing continuous variable quantum optical circuits.
Q#	Microsoft	Q#	Microsoft's quantum programming language with Visual Studio integration.

ProjectQ - IBM



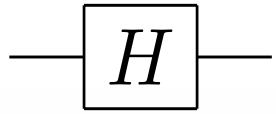
Damian S. Steiger, Thomas Häner, and Matthias Troyer, ProjectQ: An open source software framework for quantum computing. arXiv:1612.08091, 2016.

ProjectQ - Random Number Generator

```
from projectq.ops import H, Measure
from projectq import MainEngine

# create a main compiler engine
eng = MainEngine()

# allocate one qubit
q1 = eng.allocate_qubit()

# put it in superposition
H | q1

# measure
Measure | q1
eng.flush()

print("Measured: {}".format(int(q1)))
```

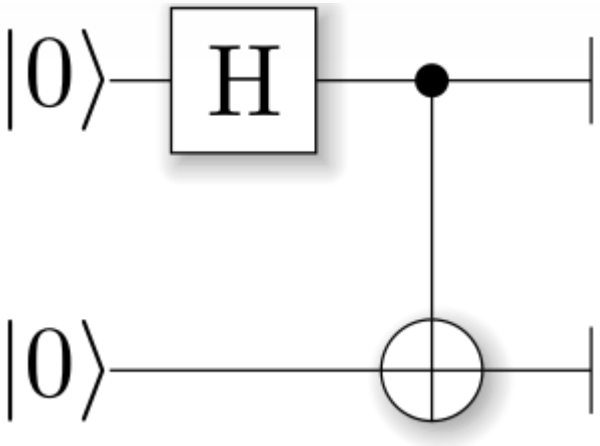
$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



ProjectQ - Bell Pair

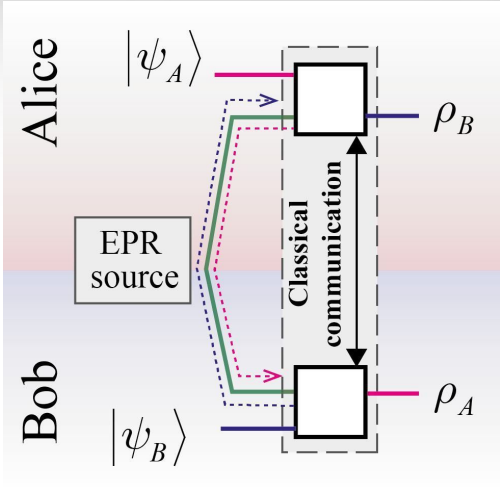
```
# create a Bell pair
def create_bell_pair(eng):
    b1 = eng.allocate_qubit()
    b2 = eng.allocate_qubit()

    H | b1
    CNOT | (b1, b2)

    return b1, b2
```



$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

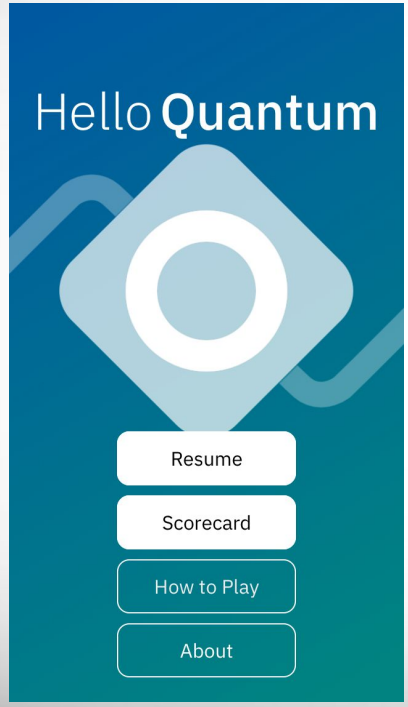
ProjectQ - Teleportation

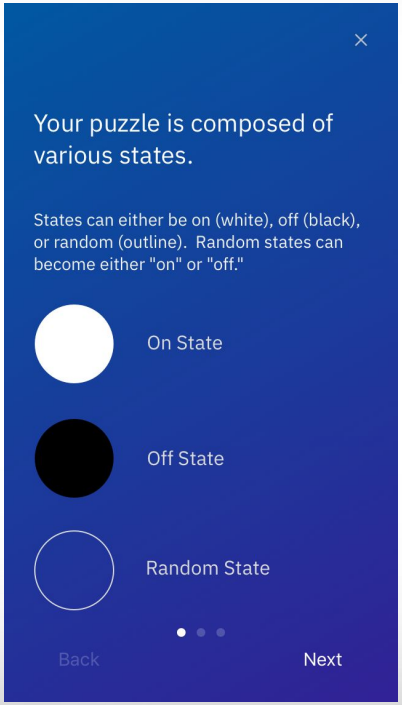
```
# make a Bell-pair
b1, b2 = create_bell_pair(eng)
# Alice creates a nice state to send
psi = eng.allocate_qubit()
if verbose:
    print("Alice is creating her state from scratch, i.e., |0>.")
state_creation_function(eng, psi)

# entangle it with Alice's b1
CNOT | (psi, b1)
if verbose:
    print("Alice entangled her qubit with her share of the Bell-pair.")
# measure two values (once in Hadamard basis) and send the bits to Bob
H | psi
Measure | psi
Measure | b1
msg_to_bob = [int(psi), int(b1)]
# Bob may have to apply up to two operation depending on the message sent
# by Alice:
with Control(eng, b1):
    X | b2
with Control(eng, psi):
     Z | b2
```

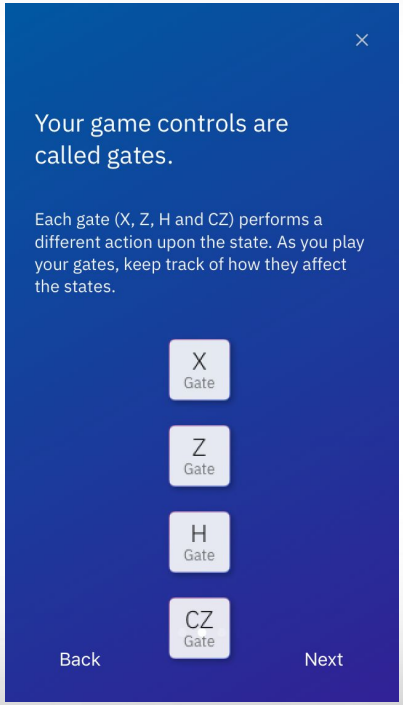


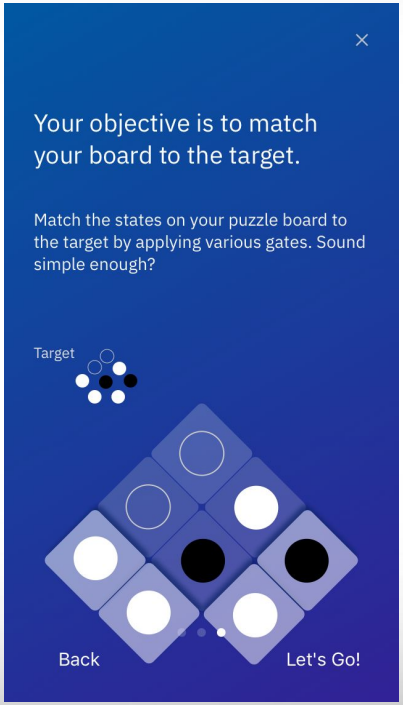
Alice has a qubit in some interesting state |ψ⟩, which she would like to show to Bob. She can do a quantum teleportation with a Bell pair.

Hello Quantum - IBM









Online Resources

Quantum Inspire

https://www.quantum-inspire.com/

An Introduction to Quantum Computing by Noson S. Yanofsky

https://arxiv.org/abs/0708.0261

Lecture notes from John Preskill at Caltech

http://www.theory.caltech.edu/~preskill/ph219/index.html

Quantum made simple

https://toutestquantique.fr/en/

Acknowledgements

- The slides are created based on the materials from ProjectQ official documentation and related Wikipedia entried.
- Supports from Texas A&M Engineering Experiment Station (TEES) and High Performance Research Computing (HPRC).