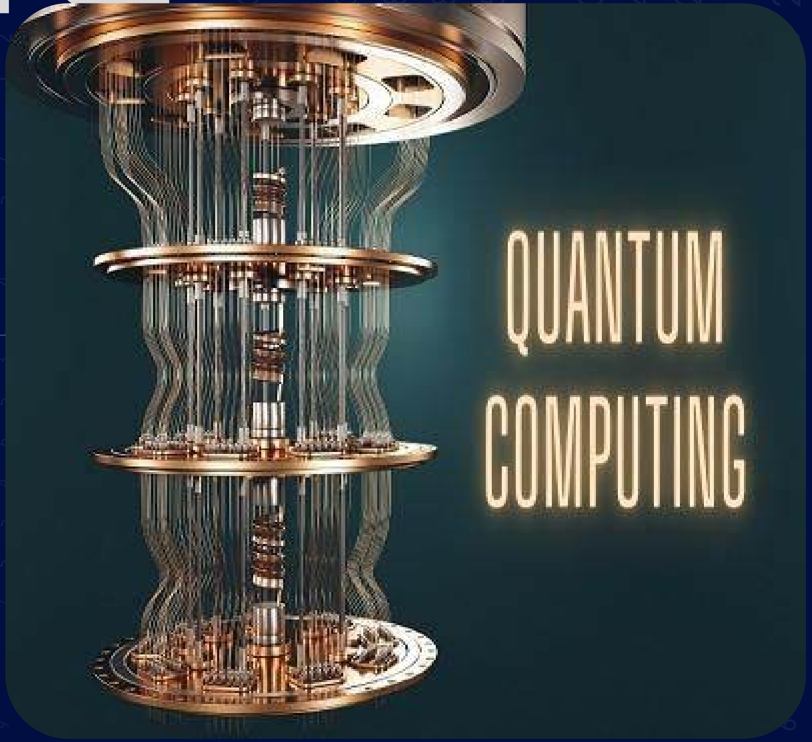
QUANTUM COMPUTING



What is QUANTUM COMPUTING ?

Quantum computing is an area of computer science that uses the principles of quantum theory. Quantum theory explains the behavior of energy and material on the atomic and subatomic levels.

Quantum computing uses subatomic particles, such as electrons or photons. Quantum computing has the capability to sift through huge numbers of possibilities and extract potential solutions to complex problems and challenges. Where classical computers store information as bits with either 0s or 1s, quantum computers use qubits

what is Quantum Computing

1 Quantum computers use quantum bits or qubits, which can exist in multiple states simultaneously due to superposition. Classical computers use classical bits that can only represent either 0 or 1.

2 Qubits can exist in a superposition of states, which means they can represent both 0 and 1 at the same time. This allows quantum computers to explore multiple possibilities in parallel.

3 Qubits can be entangled, meaning the state of one qubit is dependent on the state of another, regardless of the distance between them.

4 Quantum computers use quantum gates to manipulate qubits. These gates can perform operations on qubits in superposition, taking advantage of quantum properties to perform calculations.

5 Quantum computers use specialized quantum algorithms to solve certain problems more efficiently than classical algorithms.

Features of Quantum Computing

SUPERPOSITION

According to IBM, it's what a qubit can do rather than what it is that's remarkable. A qubit places the quantum information that it contains into a state superposition.

ENTANGLEMENT

Entanglement it is integral to quantum computing power. Pairs of qubits can be made to become entangled. This means that the two qubits then exist in a single state.

DECOHERENCE

Decoherence occurs when the quantum behavior of qubits decays. The quantum state can be disturbed instantly by vibrations temperature changes.

EXPONENTIAL SPEEDUP

Quantum computers have the potential to provide exponential speedup for specific problems. For example, they can efficiently factor large numbers.

QUANTUM COMPUTERS IN DEVELOPMENT

1

GOOGLE is spending billions of dollars to build its quantum computer by 2029. The company opened a campus in California called Google AI to help it meet this goal.

2

IBM plans to have a 1,000-qubit quantum computer in place by 2023. For now, IBM allows access to its machines for those research organizations.

3

MICROSOFT offers companies access to quantum technology via the Azure Quantum platform.

4

OTHERS There's interest in quantum computing and its technology from financial services firms such as JPMorgan Chase and Visa.

APPLICATIONS OF QUANTUM COMPUTING

Three types of Quantum computing

Quantum computing encompasses several different types of technologies and platforms that leverage the principles of quantum mechanics to perform computational tasks. Here are some of the main types of quantum computing:

Superconducting Qubits

Superconducting qubits are one of the most widely used types of qubits in quantum computing. They are tiny circuits made from superconducting materials that can carry electrical current

1

Trapped Ion Quantum Computers

Trapped ion quantum computers use ions (usually trapped in electromagnetic fields) as qubits. These qubits are manipulated using lasers. Companies like Ion and Honeywell are actively working.

2

Topological Qubits

Topological qubits are a relatively new approach to quantum computing. They are based on the concept of topological quantum states that are highly robust against errors.

3

Quantum Computing in Industry

Quantum computing has the potential to revolutionize various industries by offering the capability to solve complex problems much faster than classical computers.

FINANCE AND RISK ANALYSIS

Quantum computing optimizes portfolios pricing derivatives, enabling risk assessment.

SUPPLY CHAIN AND LOGISTICS

Quantum algorithms streamline route planning, inventory management, and demand forecasting for cost savings.

Pharmaceuticals and Drug Discovery

Quantum simulations accelerate drug discovery by modeling molecular interactions..

How do Quantum Computers work

Quantum Computers

- Superposition
- Entanglement
- Quantum Gates
- Quantum Gates
- Quantum Algorithms
- Measurement

Quantum Computing vs. Classical Computing

Quantum Computing

- Uses qubits that can exist in a superposition of states, representing both 0 and 1 simultaneously.
- Utilizes quantum gates to manipulate qubits, exploiting superposition and entanglement properties.
- Leverages superposition, allowing qubits to represent multiple states simultaneously.
- Qubits can become entangled, where the state of one qubit depends on the state of another, even when separated.

Classical Computing

- Uses classical bits (0 or 1) as the fundamental unit of information.
- Operates on classical bits using logic gates that manipulate bits based on Boolean logic.
- Does not have a superposition property; each bit is in either state 0 or 1.
- Classical bits are independent and not entangled with each other.

What are the applications of quantum computing ?

1 **Simulation**

The quantum computer is able to crunch enormous data, which includes molecular activity in one being. This ability enables the computer to create very accurate simulations of real life events, such as photosynthesis.

2 **Optimization**

Instead of calculating a problem at a time, this computing approach solves various probabilities. Through such a system, you can see how each option can provide the most advantages. Thus, you can optimize that option

3 **Cryptography**

Cryptography involves difficult mathematical problems such as discrete algorithms and integer factorization. These problems take a long time to solve. However, a computer with quantum principles can do it quickly

4 **Data search**

Although the conventional computer has evolved so much over the past decades, data search still takes too long sometimes. Quantum principles are capable of speeding up this data search process, especially if the search is unstructured.

advantages of quantum computing?

It speeds up data processing time

solving optimization problems

It can crunch a large amount of data

It is more money efficient in the long run

used to simulate quantum systems

quantum computing tools

Cirq

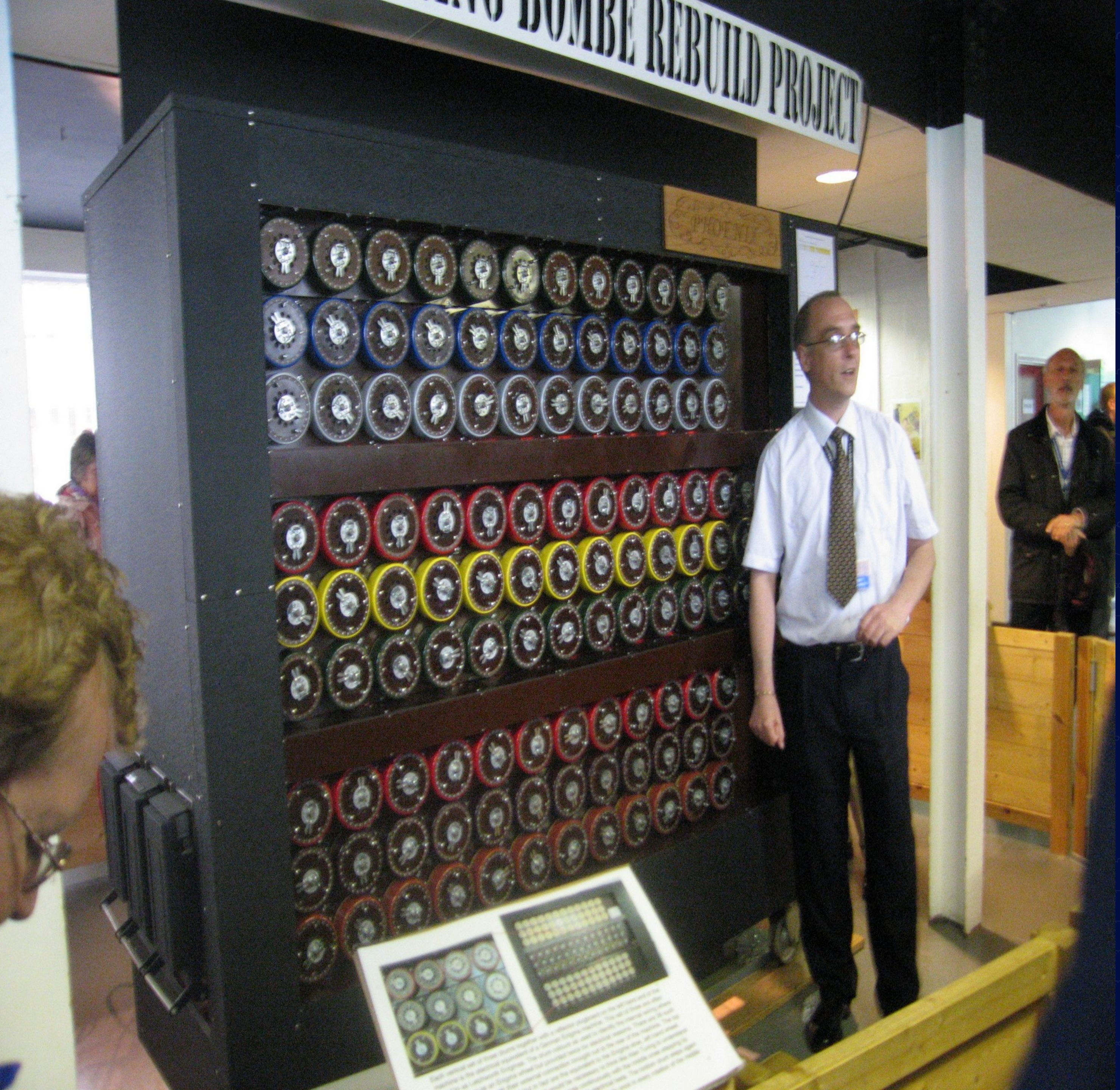
Google's Cirq is an open-source quantum programming framework that provides tools for creating, editing, and running quantum circuits. It's designed for use with Google's quantum processors.

Qiskit

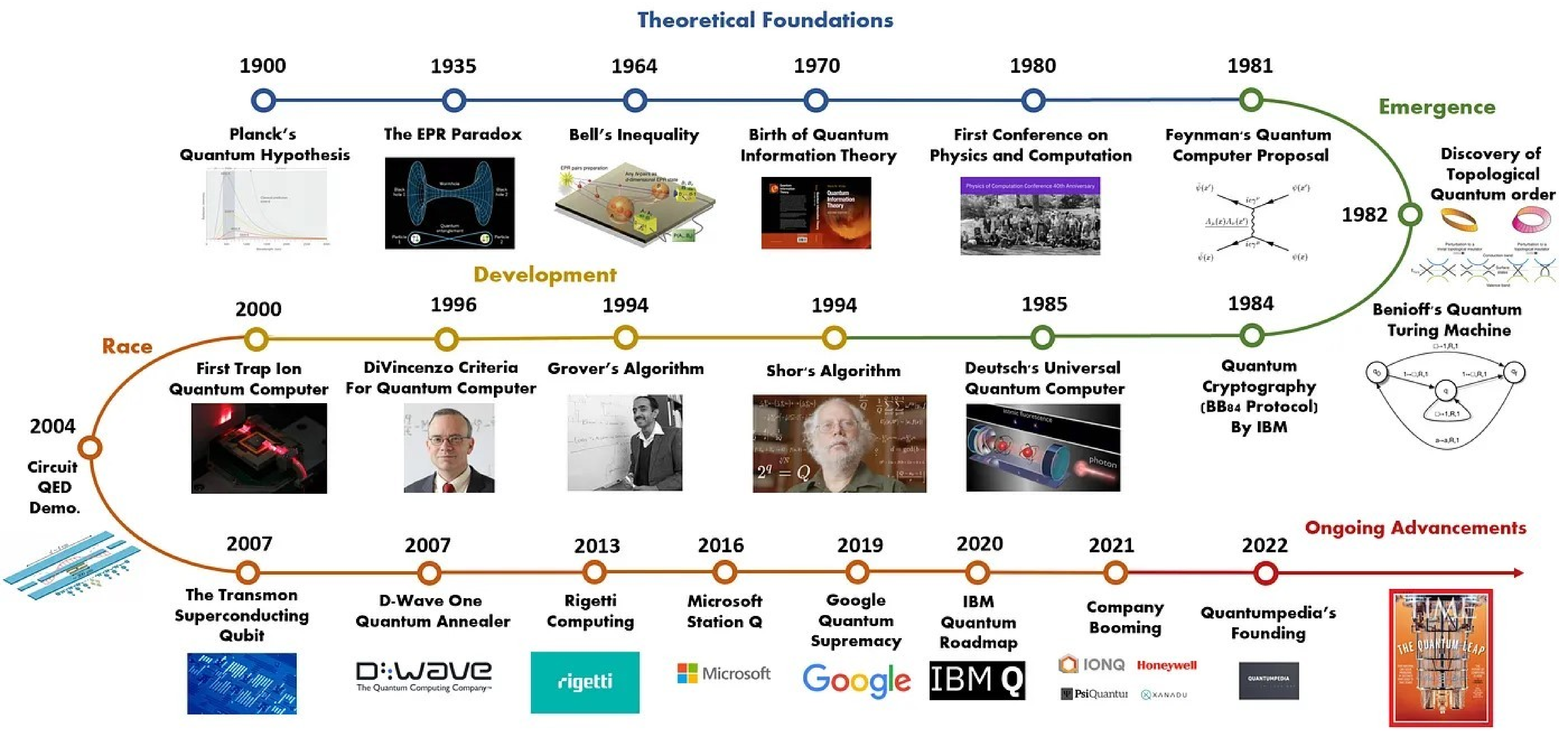
Developed by IBM, Qiskit is an open-source quantum computing framework that allows you to create and run quantum circuits, access quantum hardware, and develop quantum algorithms.

Quantum simulators

Various quantum simulators are available for simulating quantum circuits on classical hardware. These include the simulators included in Qiskit and Cirq, as standalone tools like the Quantum Development Kit from Microsoft.

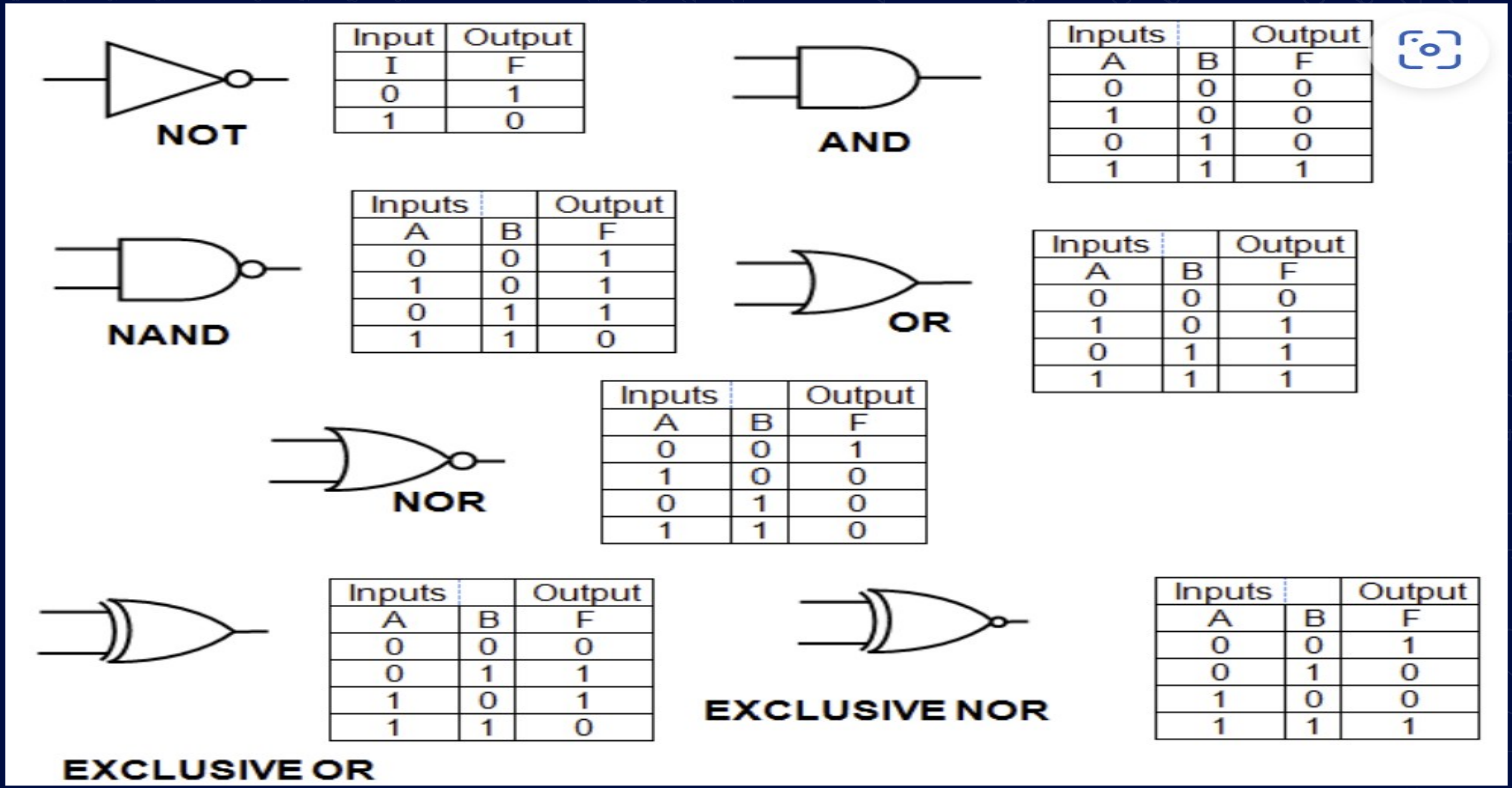


Turing machine



Theoretical Foundations
1900
Planck's Quantum Hypothesis
1935
The EPR Paradox
1964
Bell's Inequality
1970
Birth of Quantum Information Theory
1980
First Conference on Physics and Computation
1981
Feynman's Quantum Computer Proposal
Emergence
1982
Discovery of Topological Quantum order
Benioff's Quantum Turing Machine
1984
Quantum Cryptography (BB84 Protocol) By IBM
1985
Deutsch's Universal Quantum Computer
1994
Shor's Algorithm
1994
Grover's Algorithm
1996
DiVincenzo Criteria For Quantum Computer
2000
First Trap Ion Quantum Computer
Development
Race
2004
Circuit QED Demo.
2007
The Transmon Superconducting Qubit
2007
D-Wave One Quantum Annealer
D-WAVE
The Quantum Computing Company
2013
Rigetti Computing
rigetti
2016
Microsoft Station Q
Microsoft
2019
Google Quantum Supremacy
Google
2020
IBM Quantum Roadmap
IBM Q
2021
Company Booming
IONQ
Honeywell
PsiQuantum
XANADU
2022
Quantumpedia's Founding
QUANTUMPEDIA
Ongoing Advancements
TIME
THE QUANTUM LEAP

Logic gates



Input	Output
I	F
0	1
1	0

Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

Inputs		Output
A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

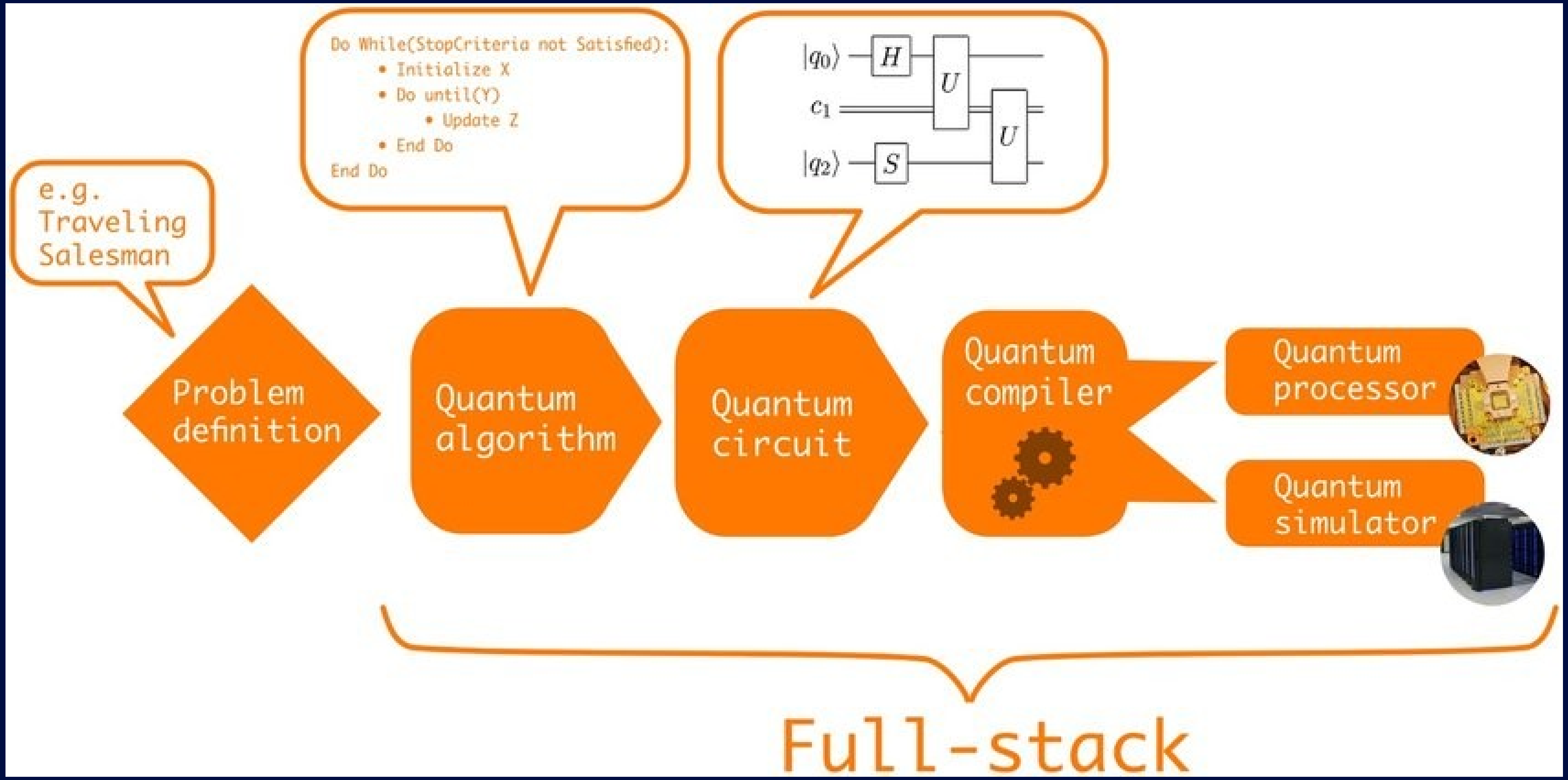
Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

Inputs		Output
A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

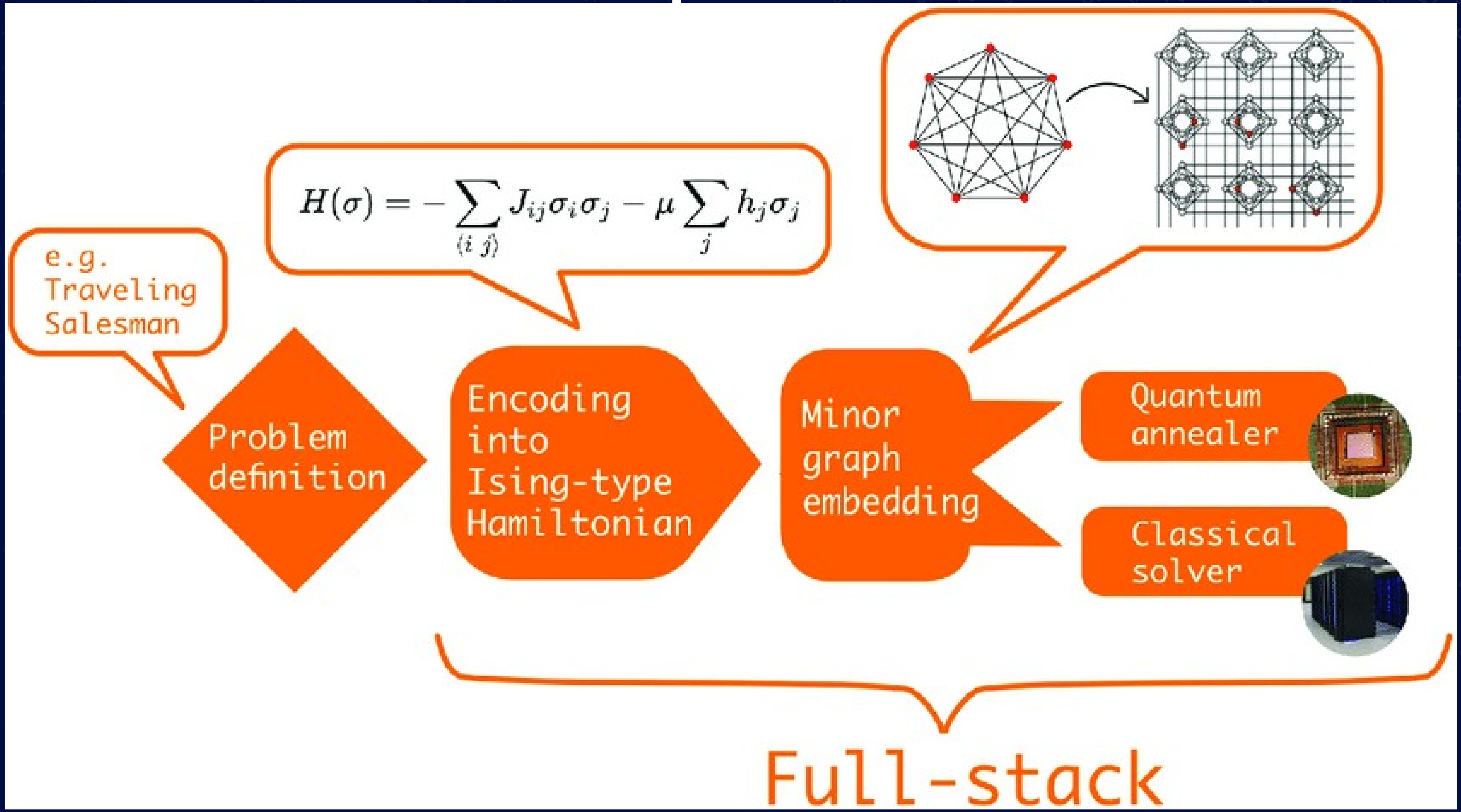
Inputs		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Inputs		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Gates Model Quantum Computation



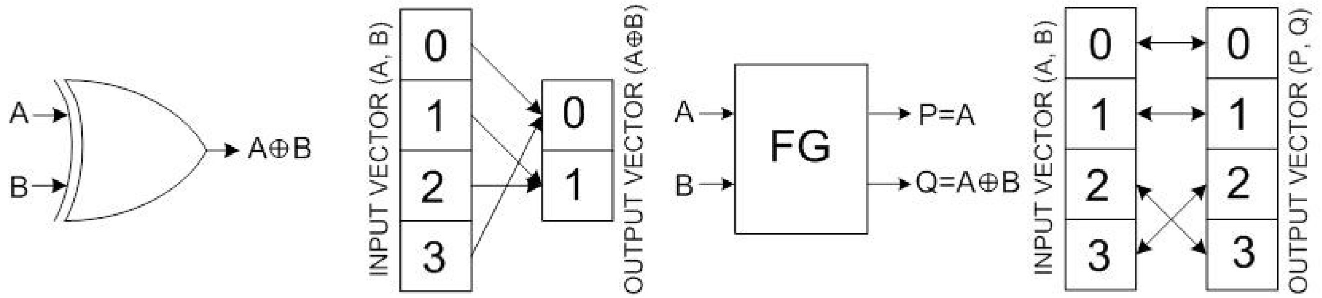
Adiabatic Model Quantum Computation



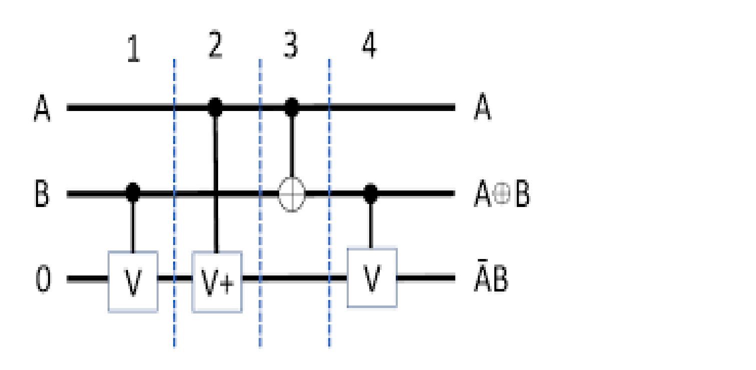
Reversible Logic (cont…)

Advantage

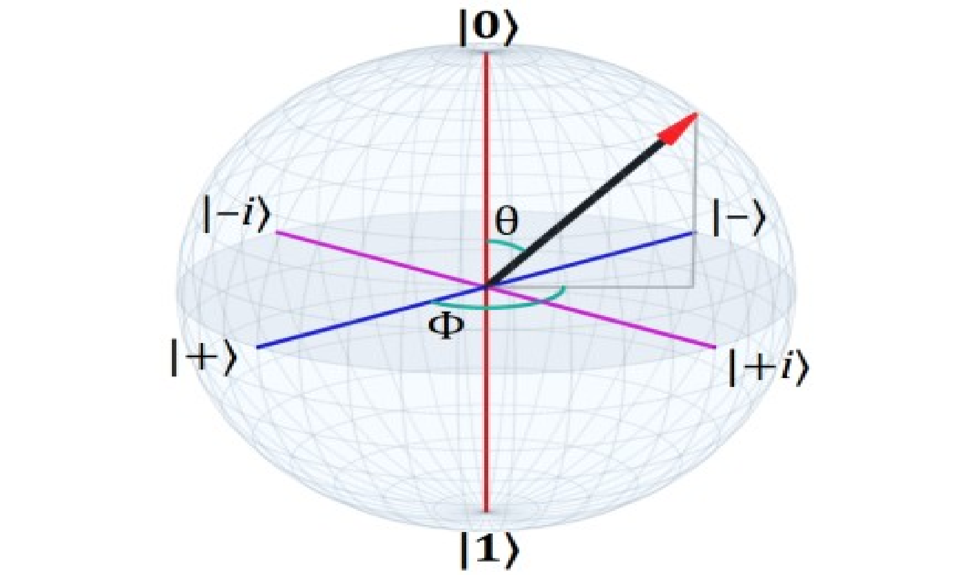
- Recovers bit-loss as well as production of heat
- Adaptable for Quantum Computing
- Multiple operations in a single cycle
- Uses low power CMOS technology

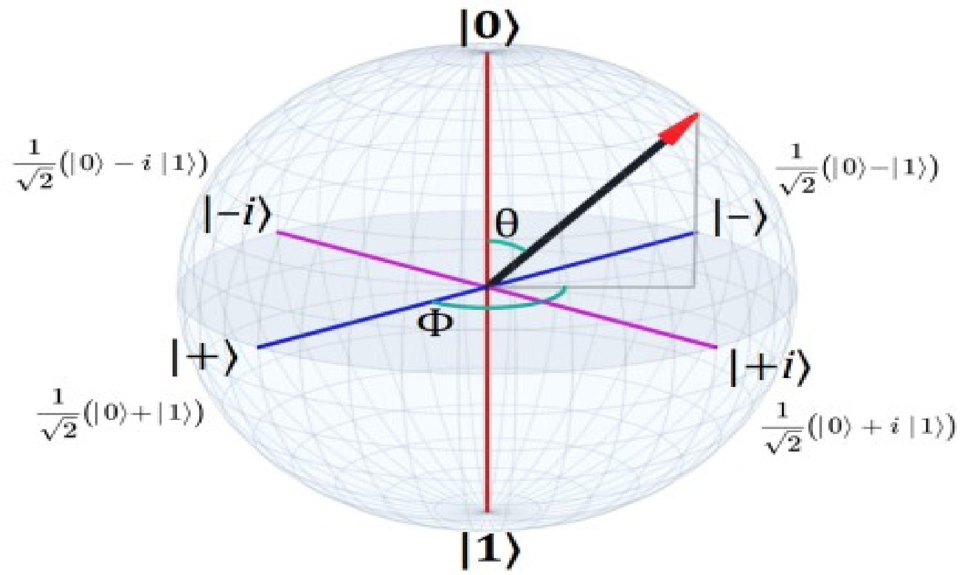


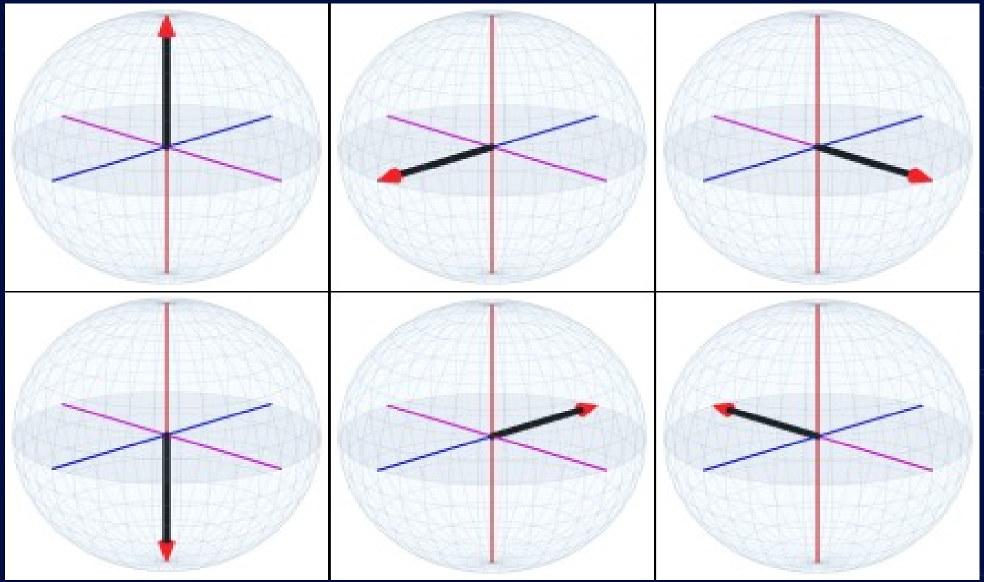
(a) Irreversible EX-OR operation (b) Reversible EX-OR operation



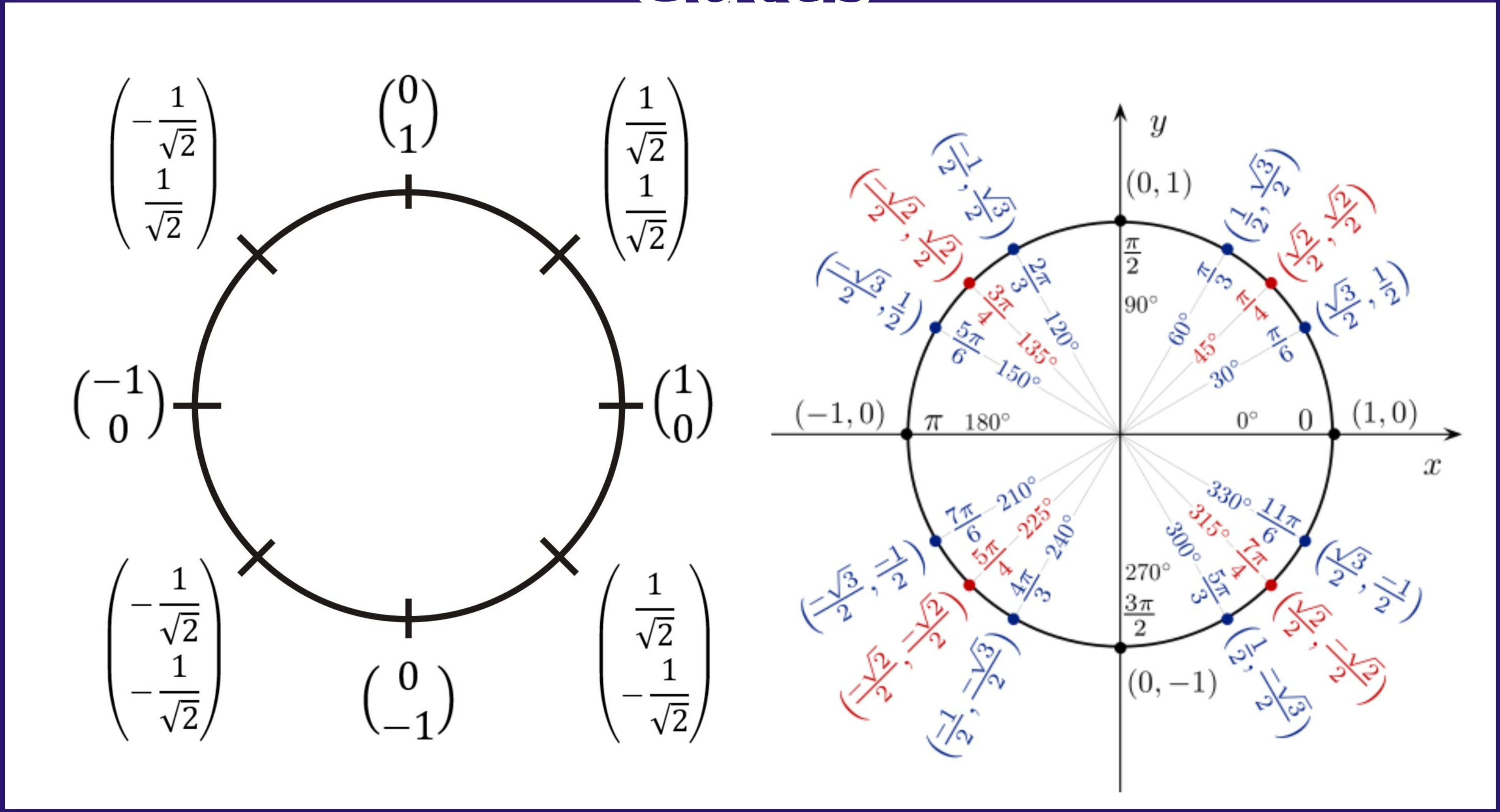
Bloch Sphere

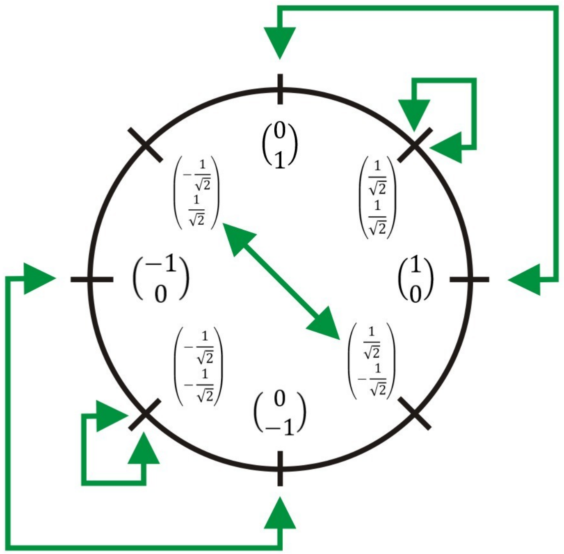


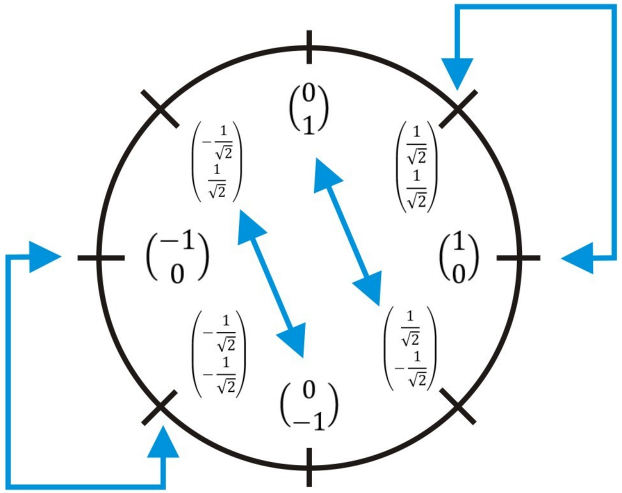




Concept Bloch Sphere for compute Gates



Gate	Input —**X**— Output
Matrix	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
State transition	
Reversibility	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$
Example	$Let\ \vert 0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \vert 1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \vert 1\rangle$

Gate	Input —**H**— Output
Matrix	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
State transition	
Reversibility	$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = I$
Example	$Let\ \vert 0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \vert 1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (\vert 0\rangle + \vert 1\rangle)$

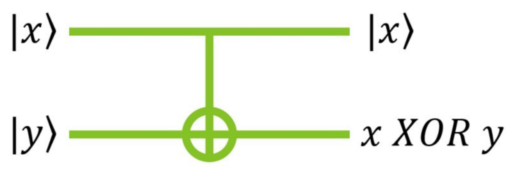
Gate	Input — **Y** — Output
Matrix	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Reversibility	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \times \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} -i^2 & 0 \\ 0 & -i^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$
Example	$Let\ \|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ i \end{bmatrix}$

Gate	Input — **Z** — Output
Matrix	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Reversibility	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$
Example	$Let\ \|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Gate	Input — **S** — Output
Matrix	$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
Reversibility	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \neq I$
Example	$Let\ \|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Gate	Input — **T** — Output
Matrix	$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$
Reversibility	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} \neq \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
Example	$Let\ \|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\ and\ \|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Gate



Matrix

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

Reversibility

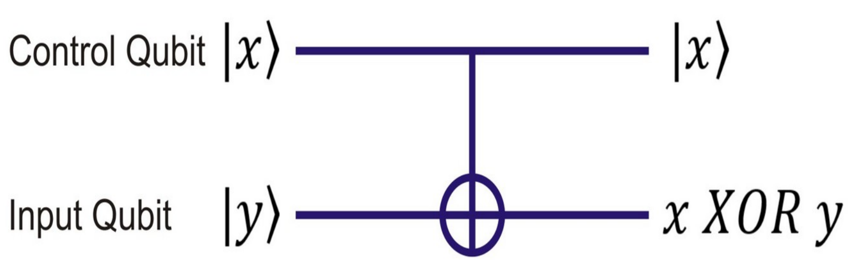
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathrm{I}$$

Example

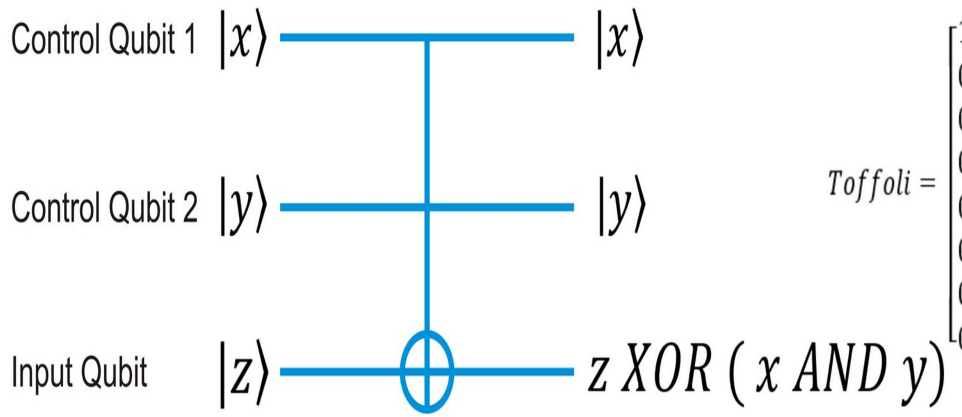
$$Let\ |x\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\ and\ |y\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|x\rangle \bigoplus |y\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \bigoplus \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

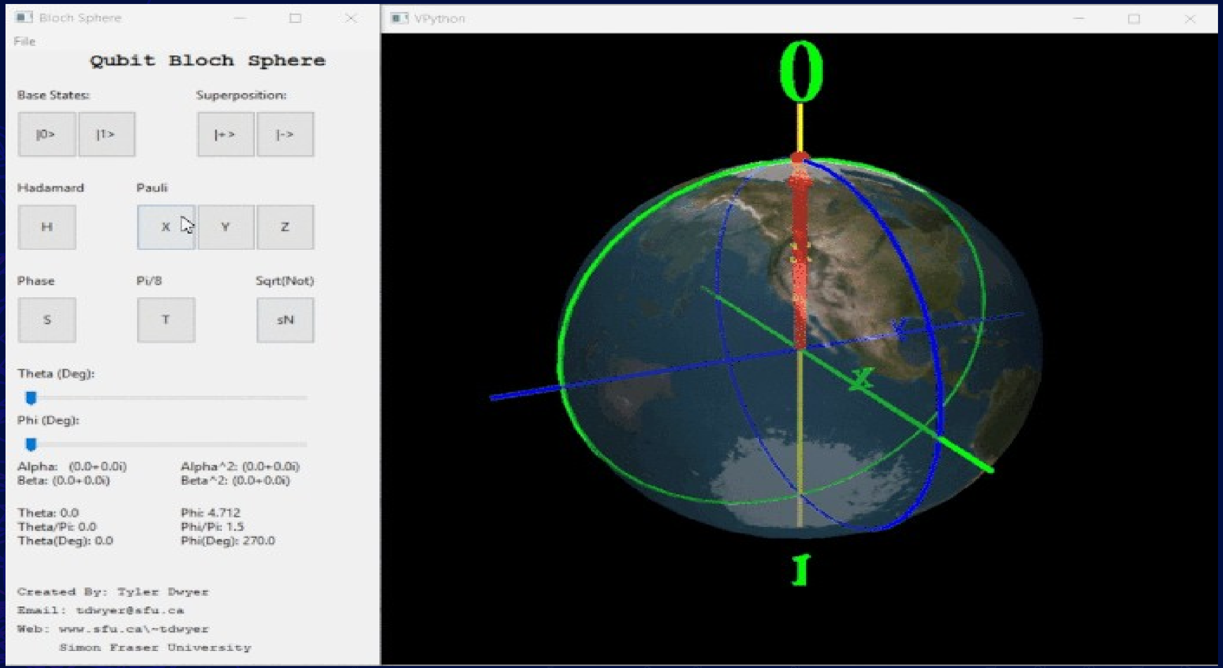
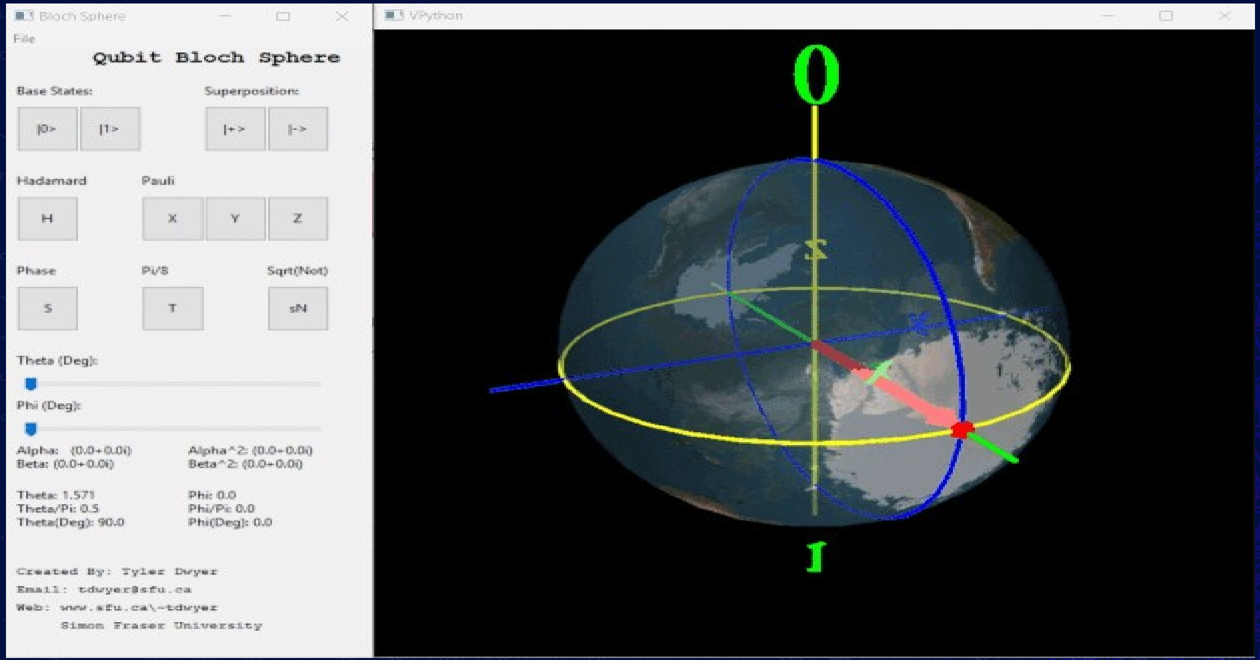
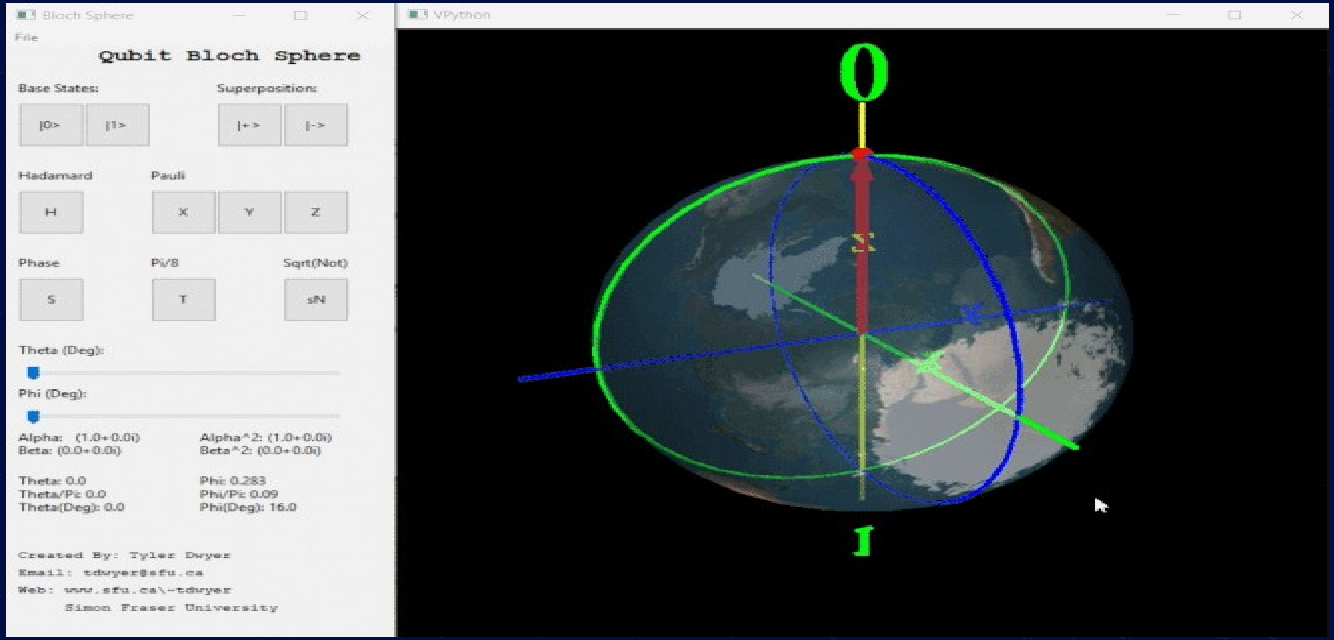
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \bigoplus \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



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



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


Bloch Sphere
VPython
File
Qubit Bloch Sphere
Base States:
Superposition:
|0>
|1>
|+>
|->
Hadamard
Pauli
H
X
Y
Z
Phase
Pi/8
Sqrt(Not)
S
T
sN
Theta (Deg):
Phi (Deg):
Alpha: (0.0+0.0i)
Beta: (0.0+0.0i)
Alpha^2: (0.0+0.0i)
Beta^2: (0.0+0.0i)
Theta: 0.0
Theta/Pi: 0.0
Theta(Deg): 0.0
Phi: 4.712
Phi/Pi: 1.5
Phi(Deg): 270.0
Created By: Tyler Dwyer
Email: tdwyer@sfu.ca
Web: www.sfu.ca\~tdwyer
Simon Fraser University
0
1

Bloch Sphere
VPython
File
Qubit Bloch Sphere
Base States:
Superposition:
|0>
|1>
|+>
|->
Hadamard
Pauli
H
X
Y
Z
Phase
Pi/8
Sqrt(Not)
S
T
sN
Theta (Deg):
Phi (Deg):
Alpha: (0.0+0.0i)
Beta: (0.0+0.0i)
Alpha^2: (0.0+0.0i)
Beta^2: (0.0+0.0i)
Theta: 1.571
Theta/Pi: 0.5
Theta(Deg): 90.0
Phi: 0.0
Phi/Pi: 0.0
Phi(Deg): 0.0
Created By: Tyler Dwyer
Email: tdwyer@sfu.ca
Web: www.sfu.ca\~tdwyer
Simon Fraser University
0
1

Bloch Sphere
VPython
File
Qubit Bloch Sphere
Base States:
Superposition:
|0>
|1>
|+>
|->
Hadamard
Pauli
H
X
Y
Z
Phase
Pi/8
Sqrt(Not)
S
T
sN
Theta (Deg):
Phi (Deg):
Alpha: (1.0+0.0i)
Beta: (0.0+0.0i)
Alpha^2: (1.0+0.0i)
Beta^2: (0.0+0.0i)
Theta: 0.0
Theta/Pi: 0.0
Theta(Deg): 0.0
Phi: 0.283
Phi/Pi: 0.09
Phi(Deg): 16.0
Created By: Tyler Dwyer
Email: tdwyer@sfu.ca
Web: www.sfu.ca\~tdwyer
Simon Fraser University
0
1

Conclusion of Gates

Gate	Transformation on Bloch sphere (defined for single qubit)
X	π-rotation around the X axis, Z→−Z. Also referred to as a bit-flip.
Z	π-rotation around the Z axis, X→−X. Also referred to as a phase-flip.
H	maps X→Z, and Z→X. This gate is required to make superpositions.
S	maps X→Y. This gate extends H to make complex superpositions. ($\pi/2$ rotation around Z axis).
$S^{\dagger}$	inverse of S. maps X→−Y. ($-\pi/2$ rotation around Z axis).
T	$\pi/4$ rotation around Z axis.
$T^{\dagger}$	$-\pi/4$ rotation around Z axis.

THANK YOU