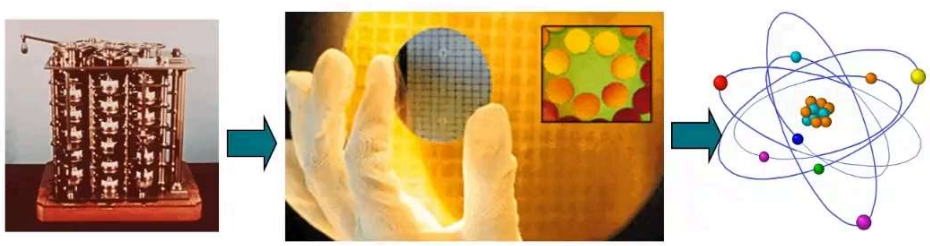
Quantum Gates & Circuits: Submodule 1
Quantum Computing using Qbits

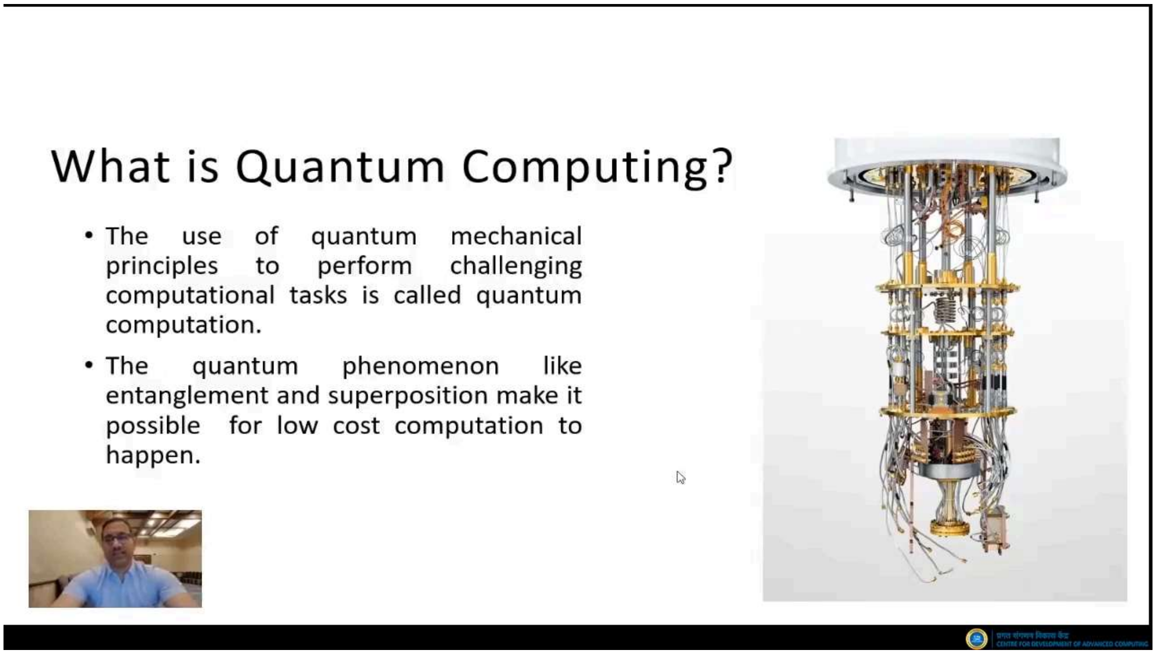
Prof. Amlan Chakrabarti
University of Calcutta
email: acakcs@caluniv.ac.in

Towards Quantum

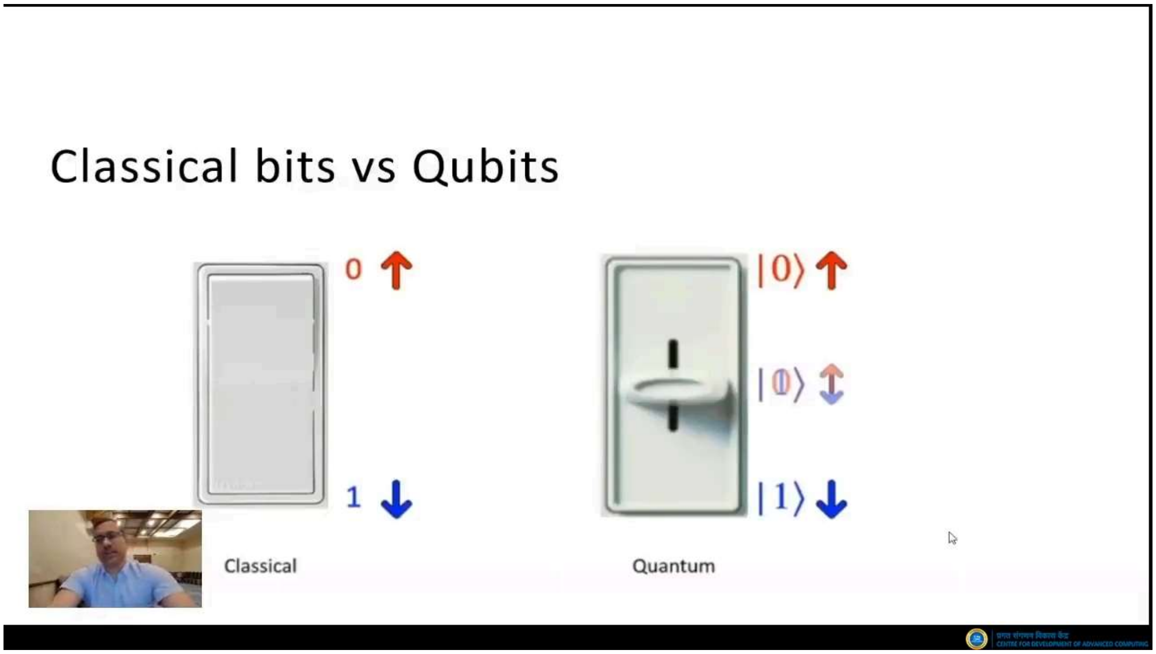


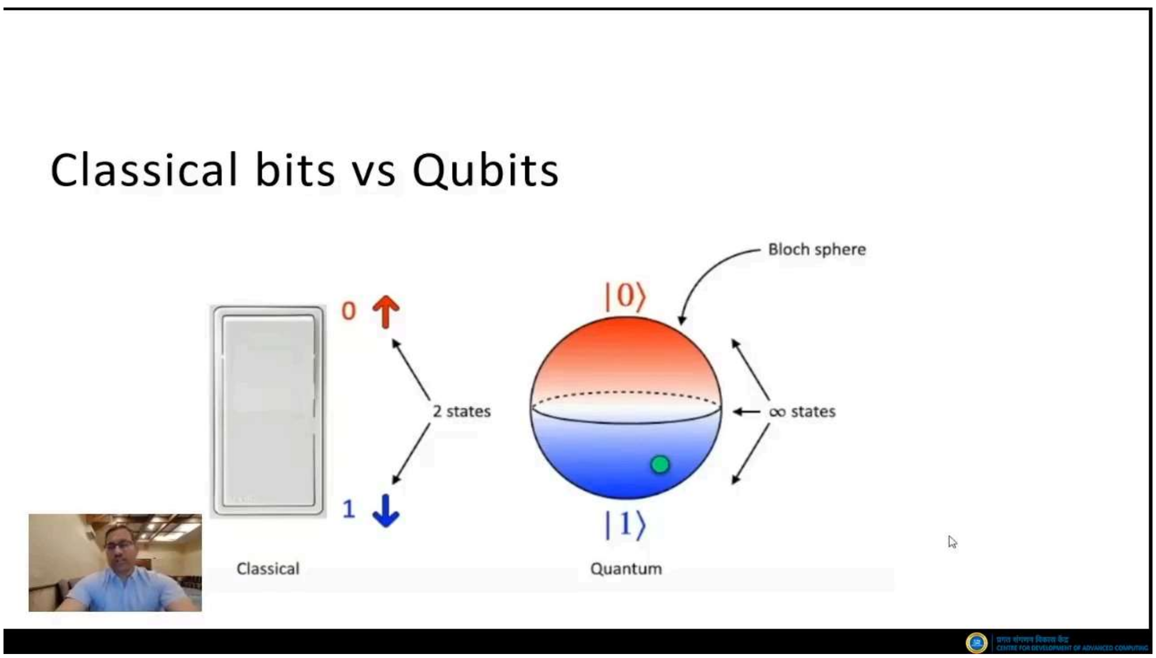
Realizations are getting smaller (and faster) and reaching a point where "classical" physics is not longer a sufficient model for the laws of physics

2



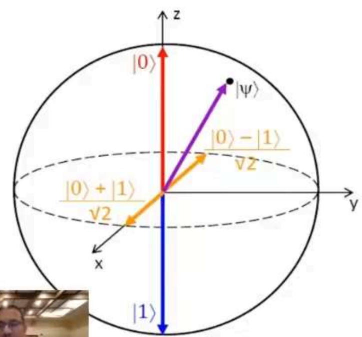
https://www.scientificamerican.com/article/ibm-releases-first-ever-1-000-qubit-quantum-chip/#:~:text=IBM%20has%20unveiled%20the%20first,error%2Dresistant%20rather%20than%20


Classical bits vs Qubits
0
1
Classical
|0⟩
|0⟩
|1⟩
Quantum


Classical bits vs Qubits
Bloch sphere
0
1
2 states
|0⟩
|1⟩
∞ states
Classical
Quantum

Computation-Transformation of the Memory State

1-qubit logic gates: rotations around x, y and z axes

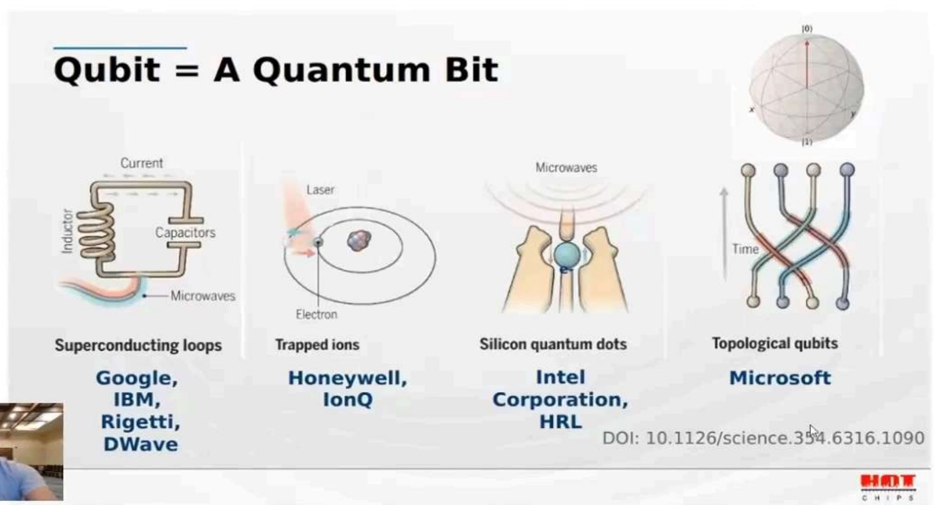


Pauli X - rotation by π around x axis
- implements NOT gate

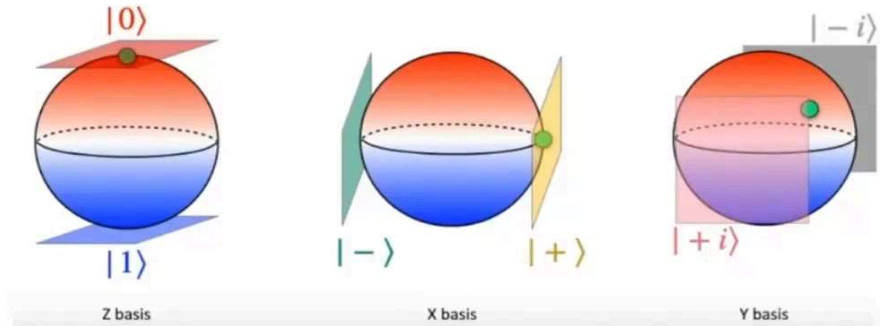
Hadamard - rotation by $\pi/2$ around y axis and then by π around x axis
- creates equal superposition when applied to $|0\rangle$ or $|1\rangle$

We can define infinitely many logic gates operating on a single qubit

Any unitary 2x2 matrix (rotation) is a logic gate



Most Important Basis for Measurement



The Postulates of Quantum Mechanics

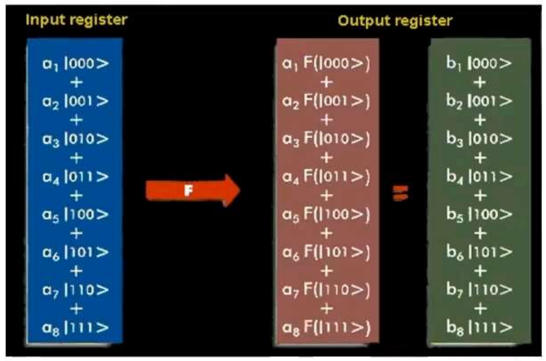
Postulate 1: Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *state vector*, which is a unit vector in the system's state space.

Postulate 2: The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2, $U|\psi\rangle = |\psi'\rangle$.

Postulate 3: Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by $p(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle$, and the state of the system after the measurement is $\frac{M_m|\psi\rangle}{\sqrt{p(m)}}$.
The measurement operators satisfy the *completeness equation*, $\sum_m M_m^\dagger M_m = I$.

The state space of a composite physical system is the tensor product of ...es of the component physical systems Moreover, if we have systems ...hrough n, and system number i, is prepared in the state $|\psi_i\rangle$, then the ... the total system is $|\psi_1\rangle \oplus |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

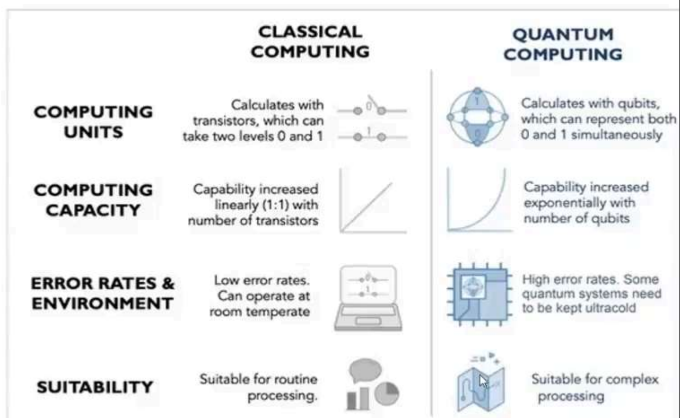
Why Is This Helpful?



- Multiple computations simultaneously
- Computing power is exponential

It’s the Scaling

- The Quantum State Space becomes 2^n large in the number of qubits, this translates a huge energy advantage.
- The energy efficiency will scale exponentially with the increase in qubits

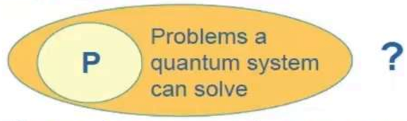


Quantum Algorithms

Richard Feynman

David Deutsch

- Feynman (1982): there may be quantum systems that cannot be simulated efficiently on a "classical" computer
- Deutsch (1985): proposed that machines using quantum processes might be able to perform computations that "classical" computers can only perform very poorly

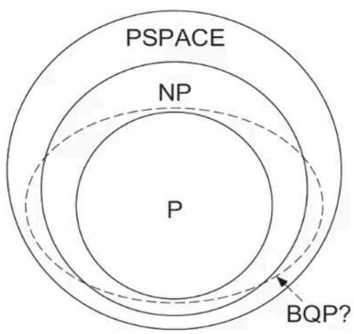


- Concept of *quantum computer* emerged as a universal device to execute such quantum algorithms

BQP (Bounded-Error Quantum Polynomial-Time): Class of problems solvable efficiently by a quantum computer

12

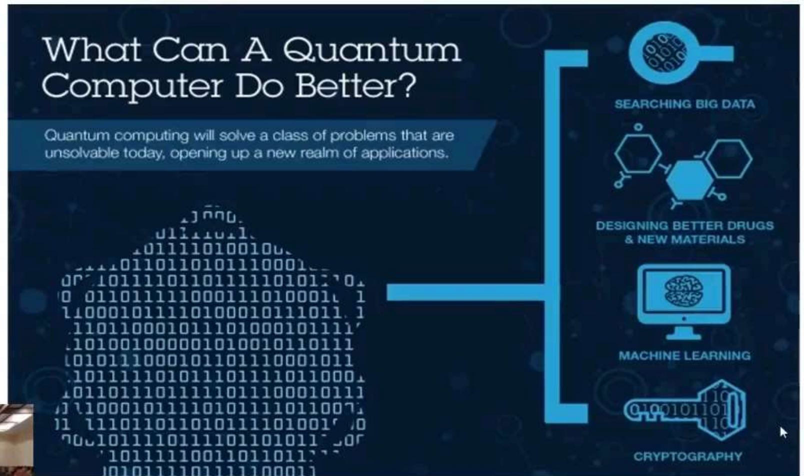
The Power of Quantum Computation



P = solved in polynomial time
NP = verified in polynomial time
PSPACE = solved in polynomial space

BQP (bounded error quantum polynomial time) is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most 1/3 for all instances.

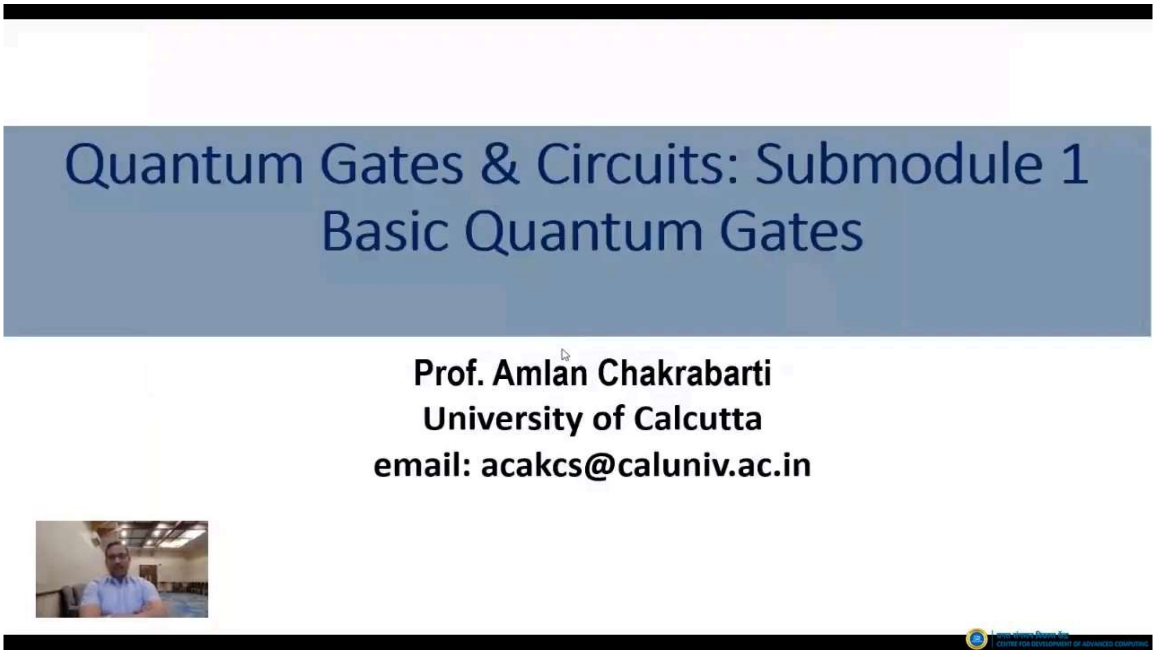
Quantum Computing Can Perform Better

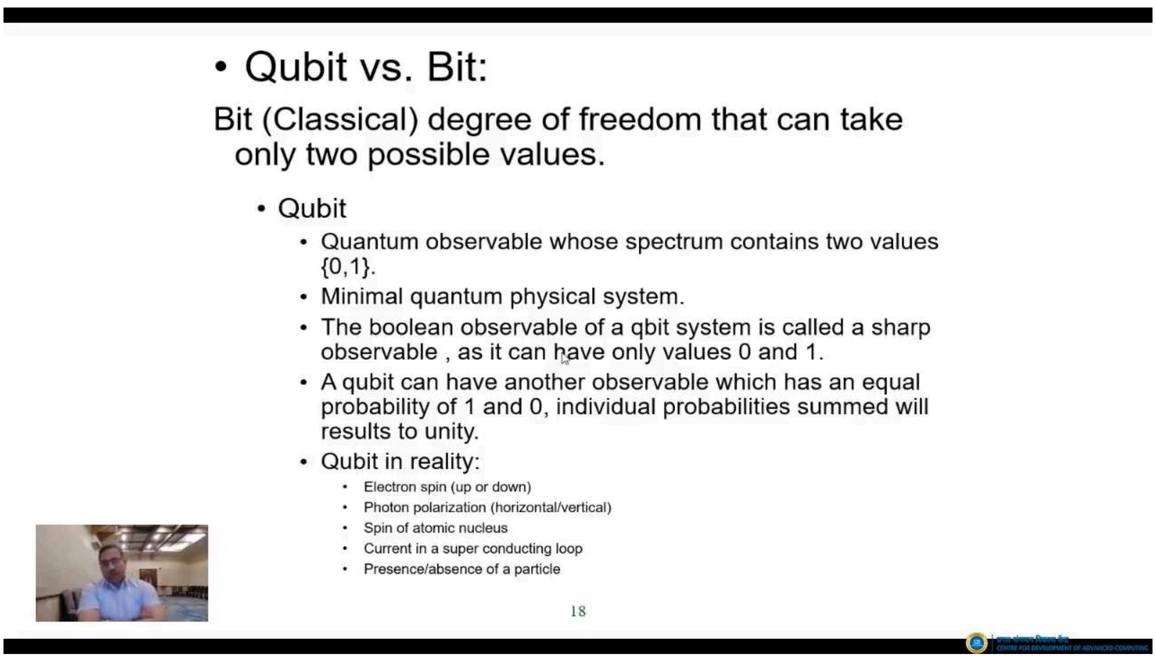


Quantum Computing: Thrust Areas

- Quantum Technology
- Quantum Algorithms
- Quantum Modelling and Simulation
- Quantum Communication and Cryptography

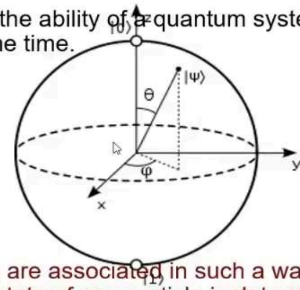
15


Quantum Gates & Circuits: Submodule 1
Basic Quantum Gates
Prof. Amlan Chakrabarti
University of Calcutta
email: acakcs@caluniv.ac.in


• Qubit vs. Bit:
Bit (Classical) degree of freedom that can take only two possible values.
• Qubit
• Quantum observable whose spectrum contains two values {0,1}.
• Minimal quantum physical system.
• The boolean observable of a qbit system is called a sharp observable , as it can have only values 0 and 1.
• A qubit can have another observable which has an equal probability of 1 and 0, individual probabilities summed will results to unity.
• Qubit in reality:
• Electron spin (up or down)
• Photon polarization (horizontal/vertical)
• Spin of atomic nucleus
• Current in a super conducting loop
• Presence/absence of a particle
18

Quantum Phenomenon: Superposition and Entanglement

- Superposition
  - Superposition is the ability of a quantum system to be in multiple states at the same time.



- Entanglement
  - Multiple particles are associated in such a way that measurement of one quantum state of one particle is determined by the measurement of the state of another particle.

19

Computation with Qubits

How does the use of qubits affect computation?

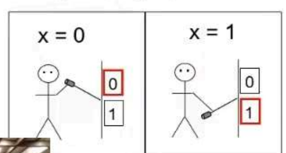
Classical Computation

Data unit: bit

● = ‘1’ ○ = ‘0’

Valid states:

x = ‘0’ or ‘1’



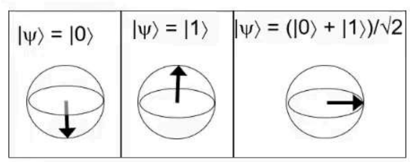
Quantum Computation

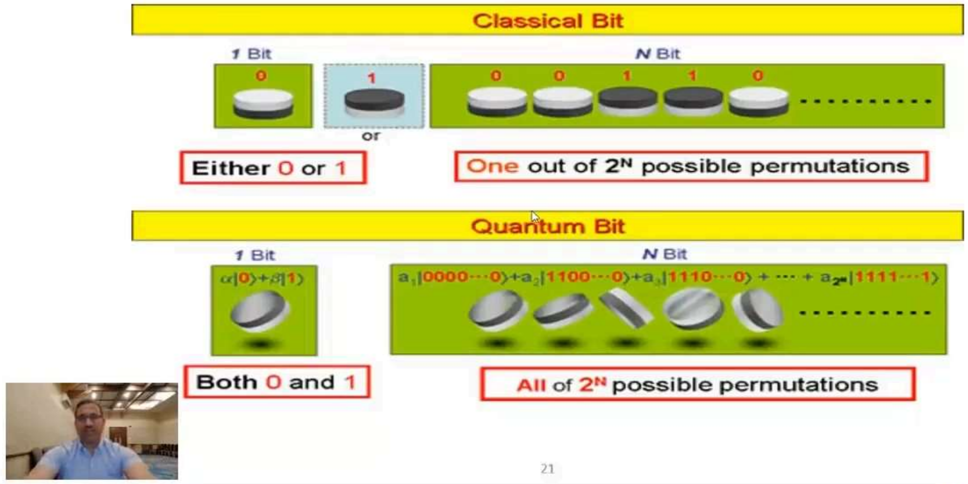
Data unit: qubit

$=|1\rangle$ $=|0\rangle$

Valid states:

$|\psi\rangle = c_1|0\rangle + c_2|1\rangle$





Entanglement

- An n-qubit system can exist in any superposition of the 2^n basis states

$$c_0|00\dots00\rangle + c_1|00\dots01\rangle + \cdots + c_{2^n-1}|11\dots11\rangle\,, \qquad \sum_{i=0}^{2^n-1} |c_i|^2 = 1$$

- If such a state can be represented as a tensor product of individual qubit states then the qubit states are **not entangled**. For example:

$$\underbrace{\left(\frac{1}{\sqrt{8}}|00\rangle + \frac{\sqrt{3}}{\sqrt{8}}|01\rangle + \frac{1}{\sqrt{8}}|10\rangle + \frac{\sqrt{3}}{\sqrt{8}}|11\rangle\right)}_{2^n \text{ probability amplitudes}} = \underbrace{\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle\right)}_{2n \text{ probability amplitudes}}$$

$$\rangle + \frac{1}{\sqrt{2}}|11\rangle\Big) \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

Quantum Logic Networks

- Invented by Deutsch (1989)
  - Analogous to classical Boolean logic networks
  - Generalization of Fredkin-Toffoli reversible logic circuits
- System is divided into individual bits, or *qubits*
  - 2 orthogonal states of each qubit are designated as the *computational basis states*, "0" and "1"
- Quantum logic gates:
  - Local unitary transforms that operate on only a few state bits at a time
- Computation via predetermined sequence of gate applications to selected bits

23

Quantum Gates: NOT

- All classical input-consuming reversible gates can be represented as unitary transformations!
- *E.g.*, input-consuming NOT gate (inverter)

in	out
0	1
1	0

$$|0\rangle :\equiv \begin{array}{c} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{array}\begin{array}{c} 0 \\ 1 \end{array} \qquad N :\equiv \begin{array}{cc} & \begin{array}{cc} 0 & 1 \end{array} \\ & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array}\begin{array}{c} 0 \\ 1 \end{array} \qquad N|0\rangle = |1\rangle$$

$$|1\rangle :\equiv \begin{array}{c} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{array}\begin{array}{c} 0 \\ 1 \end{array} \qquad N|1\rangle = |0\rangle$$

24

The Hadamard Transform

- Used frequently in quantum logic networks for generating Superpositions

$$H :\equiv \begin{array}{c} \\ \\ \end{array}\begin{array}{cc} 0 & 1 \\ \multicolumn{2}{c}{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}} \end{array}\begin{array}{c} 0 \\ 1 \end{array} \qquad H^2 = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

25

One Qbit Logic Gates

X (NOT)

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

Y

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0 & -i \\ 1 & i & 0 \end{array}$$

Z

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & -1 \end{array}$$

S

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & i \end{array}$$

T

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 0 \\ 1 & 0 & e^{i\pi/4} \end{array}$$

Hadamard (H)

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 1 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{array}$$

Sqrt NOT

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 1 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array}$$

Sqrt Root-

https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.SXGate

https://qubit.guide/2.5-the-square-root-of-not.html

Identity transformation, Pauli matrices, Hadamard

$$\delta_0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad |\varphi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

$$\delta_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad |\varphi\rangle = \alpha_1|0\rangle + \alpha_0|1\rangle$$

$$\delta_2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad |\varphi\rangle = -i\alpha_1|0\rangle + i\alpha_0|1\rangle$$

$$\delta_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad |\varphi\rangle = \alpha_0|0\rangle - \alpha_1|1\rangle$$

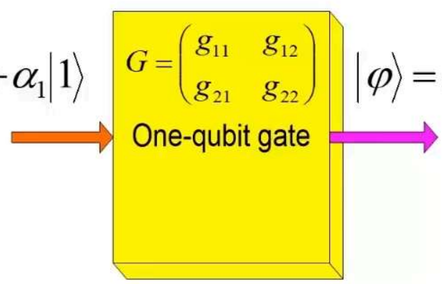
$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad |\varphi\rangle = \alpha_0\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \alpha_1\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

27

One qubit gates

- Transform an input qubit into an output qubit
- Characterized by a 2 x 2 matrix with complex coefficients

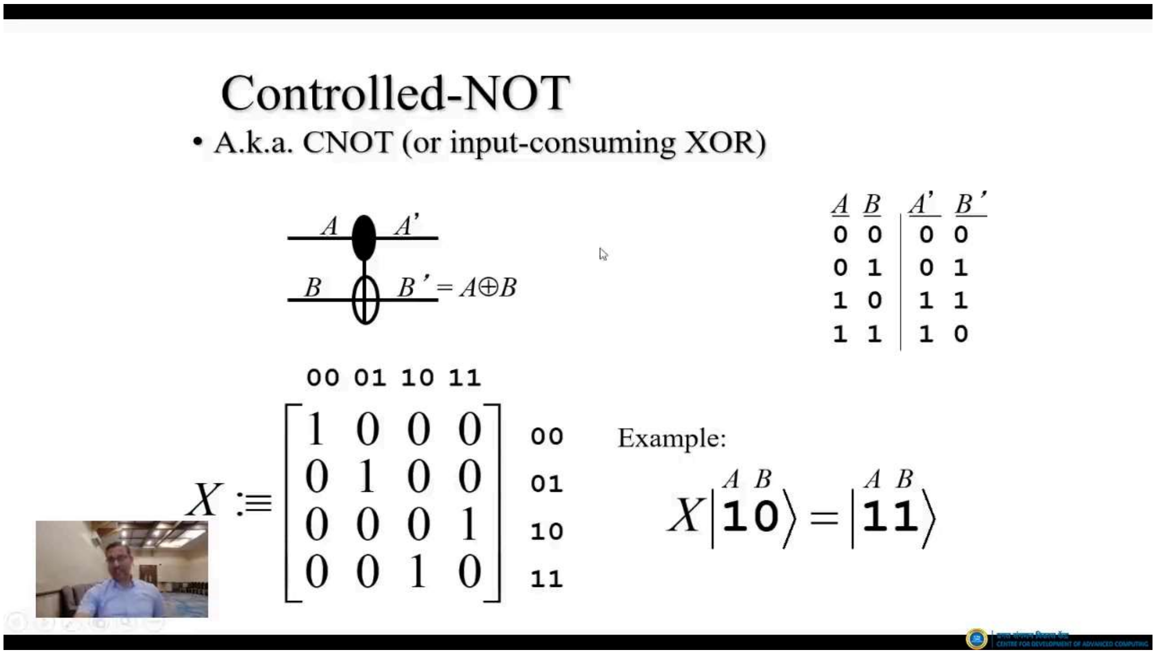
$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$



$$|\varphi\rangle = \alpha_0'|0\rangle + \alpha_1'|1\rangle$$

$$|\varphi\rangle = G|\psi\rangle \qquad \begin{pmatrix} \alpha_0' \\ \alpha_1' \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

28



In a CNOT (Controlled-NOT) gate in quantum computing, measuring one qubit can indeed provide information about the other qubit, depending on the state of the measured qubit and the entanglement between the qubits.

Explanation:

1. Entanglement in CNOT Gate:

In a CNOT gate, two qubits are involved: a control qubit (usually denoted as q_{control}) and a target qubit (usually denoted as q_{target}).

If the control qubit is in the state $|1\rangle$, it flips the state of the target qubit. If the control qubit is in the state $|0\rangle$, the target qubit remains unchanged.

2. Measurement and Entanglement:

When two qubits are entangled, measuring one qubit can instantaneously affect the state of the other qubit.

If we measure the control qubit, and it collapses to $|0\rangle$ or $|1\rangle$, we instantly know the state of the target qubit, as it will be flipped or remain the same, respectively.

3. Example:

Let's say we have a CNOT gate with qubit q_{control} as the control qubit and q_{target} as the target qubit.

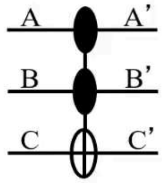
If q_{control} is in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and q_{target} is in the state $|0\rangle$ before measurement.

If we measure q_{control} and it collapses to $|0\rangle$, then q_{target} remains $|0\rangle$. If it collapses to $|1\rangle$, then q_{target} flips to $|1\rangle$.

Conclusion:

In summary, measuring one qubit in a CNOT gate can provide information about the state of the other qubit due to their entanglement. This property is fundamental in quantum computing and can be used for various quantum information processing tasks, including quantum teleportation and quantum error correction.

Toffoli Gate (CCNOT)



A	B	C		A'	B'	C'
0	0	0	→	0	0	0
0	0	1	→	0	0	1
0	1	0	→	0	1	0
0	1	1	→	0	1	1
1	0	0	→	1	0	0
1	0	1	→	1	0	1
1	1	0		1	1	0
1	1	1		1	1	1

$$
X :\equiv \begin{array}{c c}
\begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} & \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} & \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array}
\end{array}
$$

30

Computation with Qubits

How does the use of qubits affect computation?

Classical Computation

Operations: logical

Valid operations:

1-bit

NOT =

	in: 0	1
out	1	0

2-bit

AND =

in \ in	0	1
0	0	0
1	0	1

out

Quantum Computation

Operations: unitary

Valid operations:

1-qubit

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

$\sigma_Y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$ $H_d = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

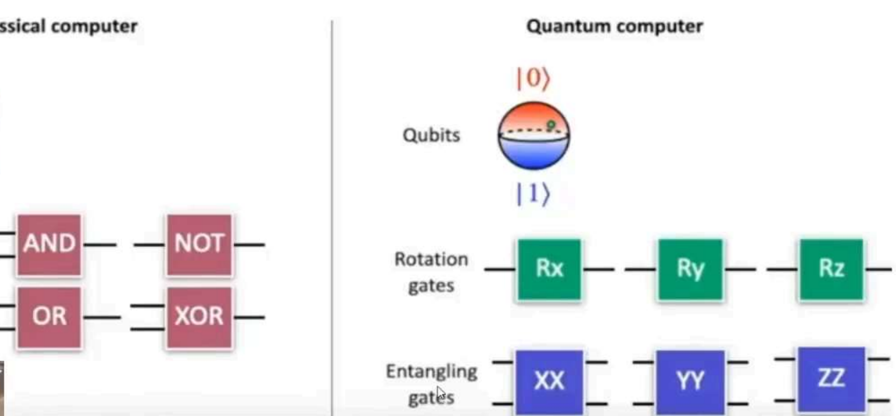
2-qubit

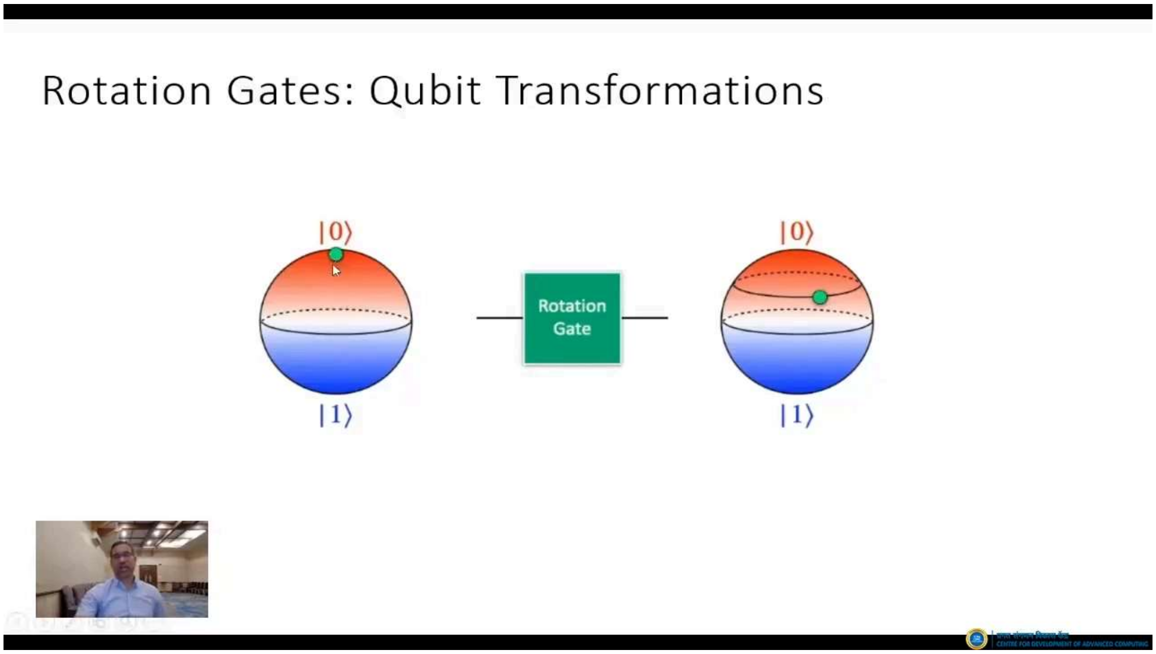
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

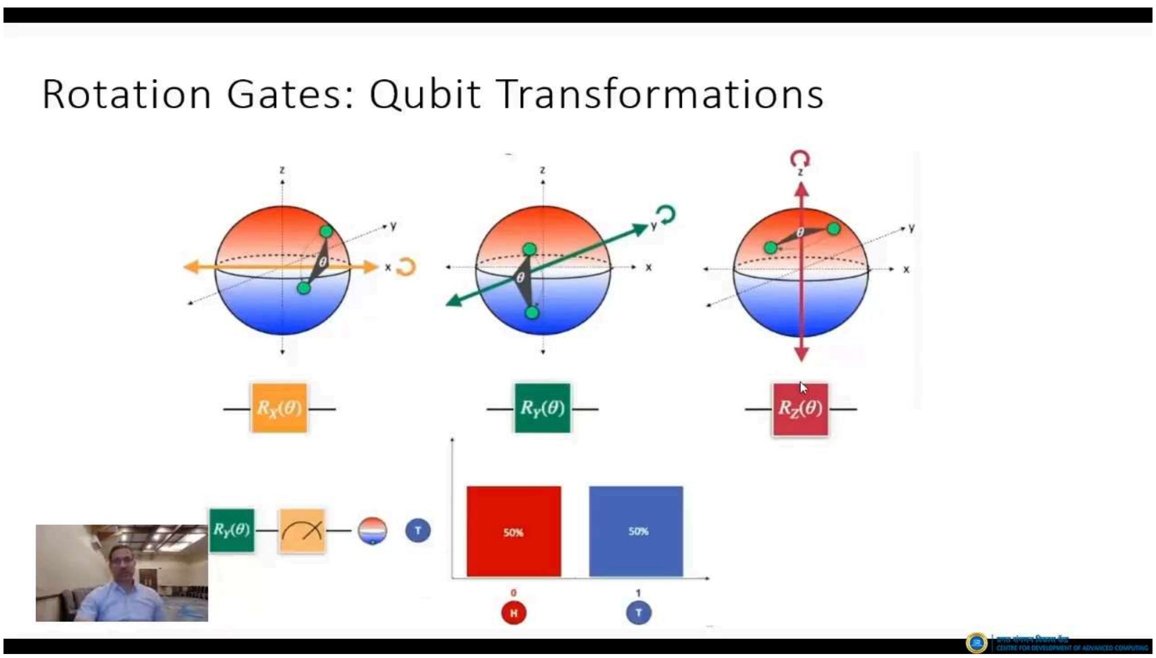
More than one qubit

	Single qubit	Two qubits
Hilbert space	$\lvert 0\rangle, \lvert 1\rangle$ $\mathcal{H}_2 = \left\{ \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix} \right\}$	$\lvert 00\rangle, \lvert 01\rangle, \lvert 10\rangle, \lvert 11\rangle$ $\mathcal{H}_2^{\otimes 2} = \mathcal{H}_2 \otimes \mathcal{H}_2 = \left\{ \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \right\}$
Arbitrary state	$\lvert\psi\rangle = c_1\lvert 0\rangle + c_2\lvert 1\rangle = \begin{bmatrix}c_1\\c_2\end{bmatrix}$	$\lvert\Psi\rangle = c_1\lvert 00\rangle + c_2\lvert 01\rangle + c_3\lvert 10\rangle + c_4\lvert 11\rangle = \begin{bmatrix}c_1\\c_2\\c_3\\c_4\end{bmatrix}$
Operator	$U\lvert\psi\rangle = \begin{pmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{pmatrix}\begin{pmatrix}c_1\\c_2\end{pmatrix}$	$U\lvert\Psi\rangle = \begin{pmatrix}u_{11} & u_{12} & u_{13} & u_{14}\\ u_{21} & u_{22} & u_{23} & u_{24}\\ u_{31} & u_{32} & u_{33} & u_{34}\\ u_{41} & u_{42} & u_{43} & u_{44}\end{pmatrix}\begin{pmatrix}c_1\\c_2\\c_3\\c_4\end{pmatrix}$

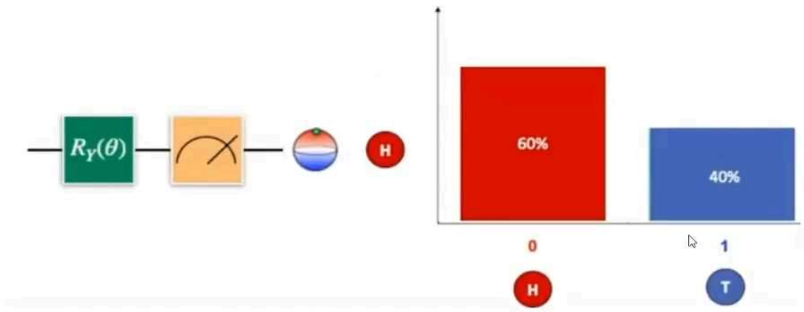
Bits, Qubits and Gates




Rotation Gates: Qubit Transformations
|0⟩
Rotation Gate
|1⟩


Rotation Gates: Qubit Transformations
$R_X(\theta)$
$R_Y(\theta)$
$R_Z(\theta)$
50%
50%
0
1

Rotation with Bias



Physical Machine Descriptions (PMDs)

- Technology for a given quantum implementation
- Quantum gate implementation is achieved by means of supported quantum operations
- Variability among the PMDs
  - primitive quantum operations
  - related cost
- Commonly known PMDs
  - Quantum dot (QD)
  - Superconducting (SC)
  - Ion trap (IT)
  - Neutral atom (NA)
  - Linear photonics (LP)
  - Nonlinear photonics (NP)

Minimal Set of 1-qubit gates

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = P(\frac{\pi}{2})R_x(\pi) = iR_x(\pi)$$

$$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = P(\frac{\pi}{2})R_y(\pi) = iR_y(\pi)$$

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = P(\frac{\pi}{2})R_z(\pi) = iR_z(\pi)$$

- Any two of *{Rx,Ry,Rz} can be converted into the third one*

$$R_x(\theta) = R_z(-\frac{\pi}{2}) \cdot R_y(\theta) \cdot R_z(\frac{\pi}{2})$$

- <u>Quantum Logic Circuits</u>
  - Circuit behavior is governed explicitly by quantum mechanics
  - Signal states are vectors interpreted as a superposition of binary "qubit" vectors with complex-number coefficients

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} c_i |i_{n-1} i_{n-1} \dots i_0\rangle$$

  - Operations are defined by linear algebra over Hilbert Space and can be represented by unitary matrices with complex elements
  - Severe restrictions exist on copying and measuring signals
  - Many universal gate sets exist but the best types are not obvious

40

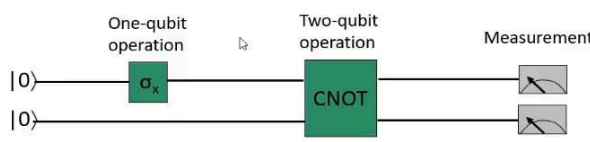
Quantum Circuit Characteristics

- Unitary Operations
  - Gates and circuits must be reversible (information-lossless)
    - Number of output signal lines = Number of input signal lines
    - The circuit function must be a bijection, implying that output vectors are a permutation of the input vectors
  - Classical logic behavior can be represented by permutation matrices
  - Non-classical logic behavior can be represented including state sign (phase) and entanglement

41

Quantum Circuit Model

Example Circuit

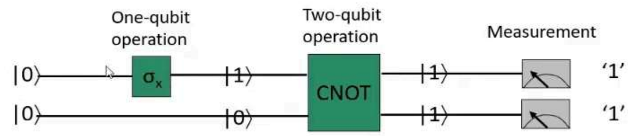


$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$\sigma_x \otimes I = \begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix} \qquad \text{CNOT} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

Quantum Circuit Model

Example Circuit

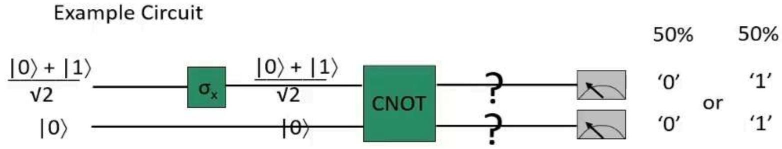


$$\begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

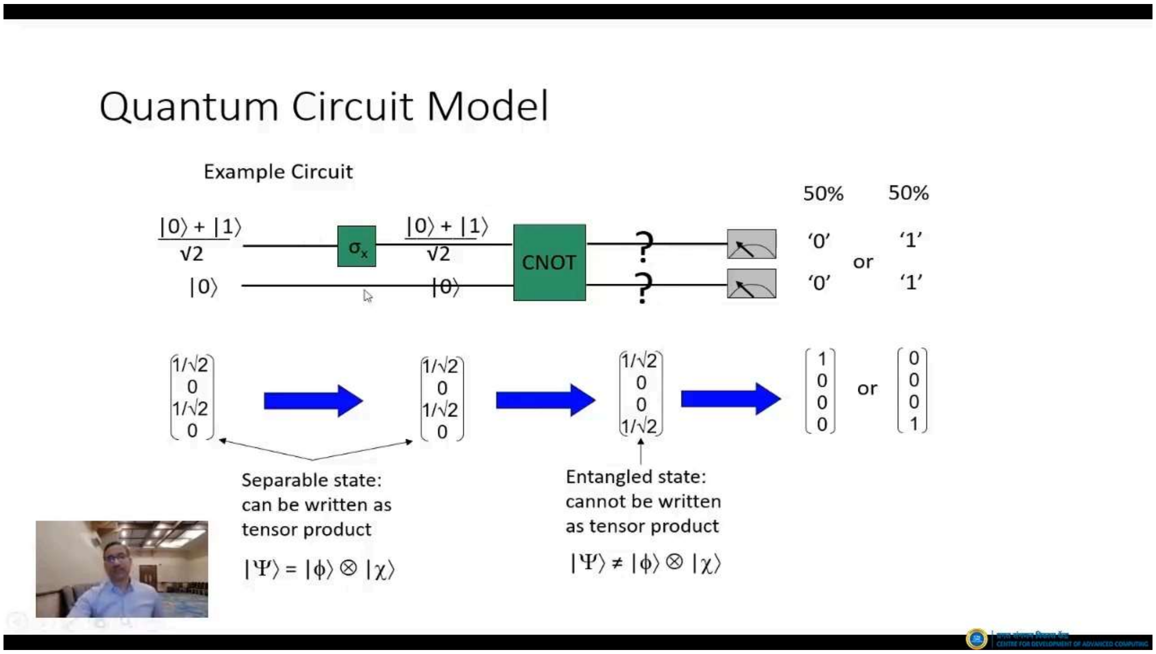
$$\sigma_x \otimes I = \begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix} \qquad CNOT = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}$$

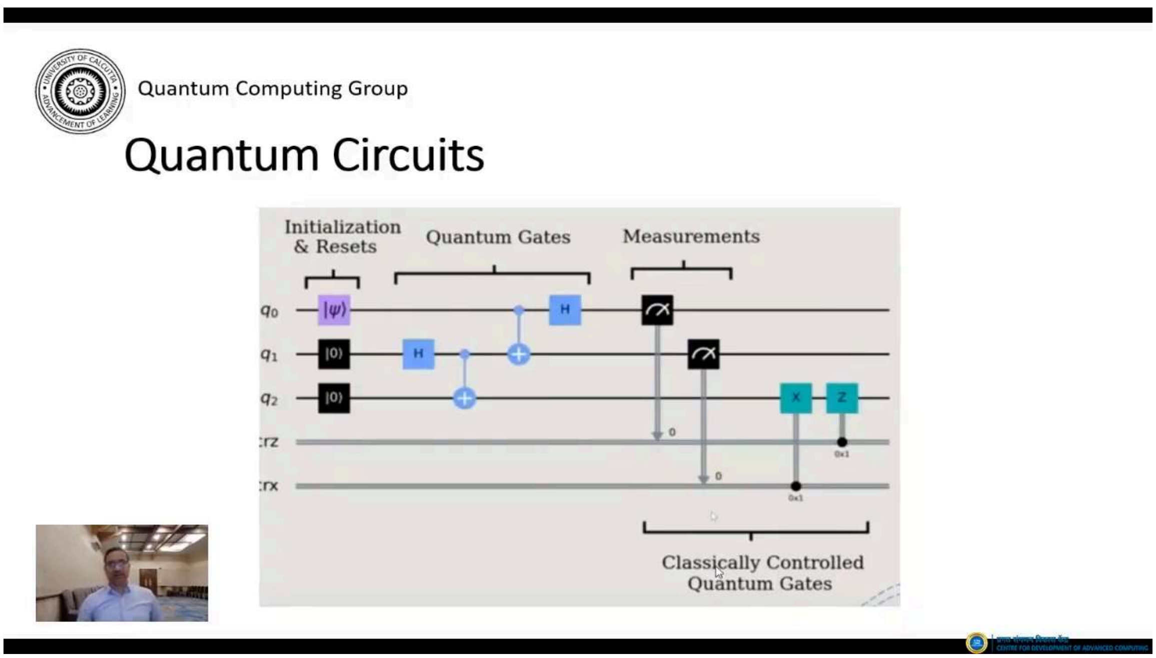
Quantum Circuit Model

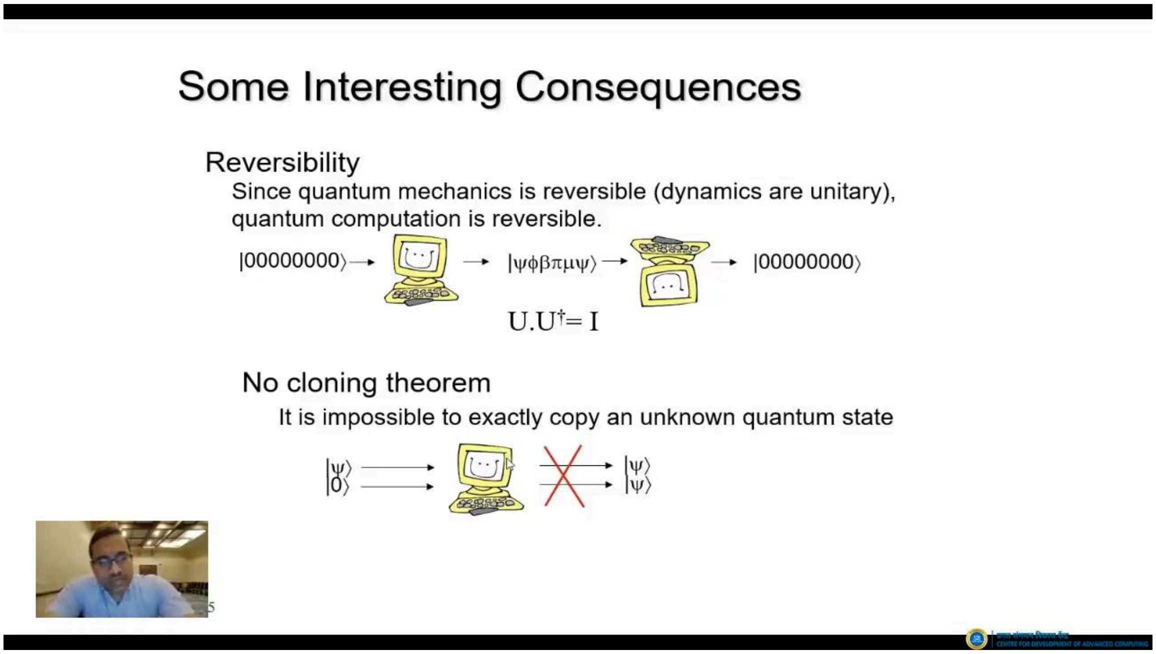
Example Circuit



$$\begin{pmatrix}1/\sqrt{2}\\0\\1/\sqrt{2}\\0\end{pmatrix} \rightarrow \begin{pmatrix}1/\sqrt{2}\\0\\1/\sqrt{2}\\0\end{pmatrix} \rightarrow \begin{pmatrix}1/\sqrt{2}\\0\\0\\1/\sqrt{2}\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \text{ or } \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$


Quantum Circuit Model
Example Circuit
50%
50%
|0⟩ + |1⟩ / √2
|0⟩
σx
CNOT
'0'
'0'
or
'1'
'1'
Separable state:
can be written as
tensor product
|Ψ⟩ = |ϕ⟩ ⊗ |χ⟩
Entangled state:
cannot be written
as tensor product
|Ψ⟩ ≠ |ϕ⟩ ⊗ |χ⟩


Quantum Computing Group
Quantum Circuits
Initialization & Resets
Quantum Gates
Measurements
Classically Controlled Quantum Gates



The No-Cloning Theorem is a fundamental concept in quantum mechanics that states it is impossible to create an exact copy of an arbitrary unknown quantum state. In classical physics, copying information is straightforward; you can duplicate a document, a file, or any piece of information without altering the original.

However, in quantum mechanics, the situation is different due to the principles of superposition and entanglement. Here's why the No-Cloning Theorem holds:

1. **Superposition**: Quantum states can exist in a superposition of multiple states simultaneously. For example, a qubit can be in a state like $ \frac{1}{\sqrt{2}} \left(\vert 0 \rangle + \vert 1 \rangle \right) $, representing both 0 and 1 with certain probabilities.

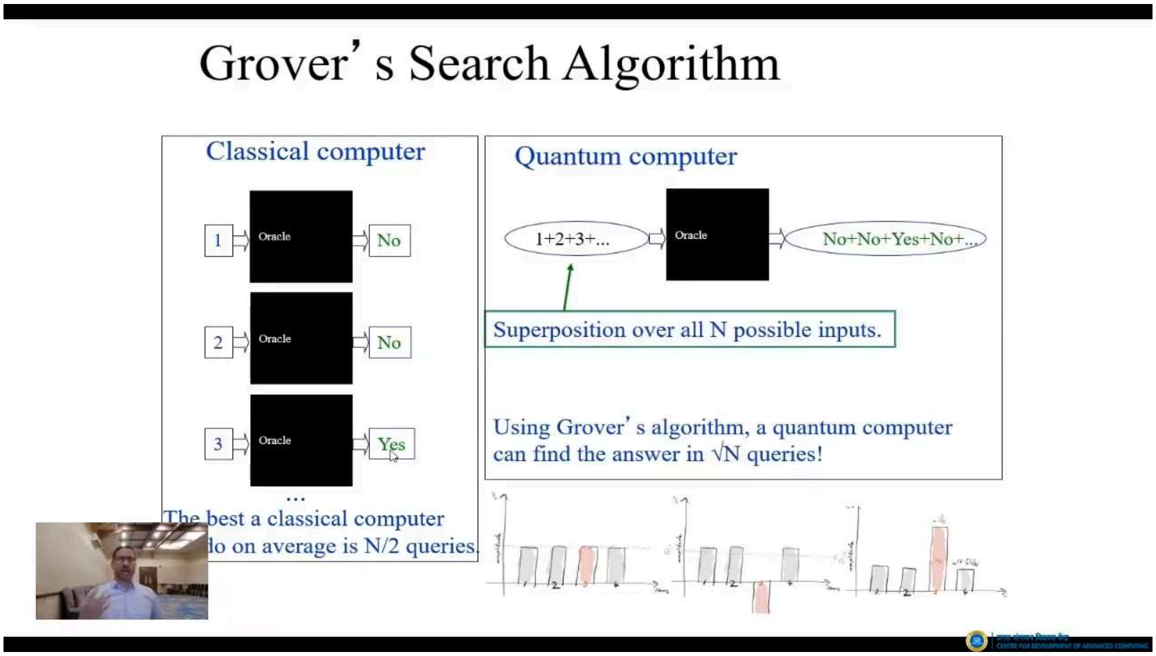
2. **Measurement**: When you measure a quantum state, it "collapses" into one of its possible states based on the probabilities dictated by its superposition.

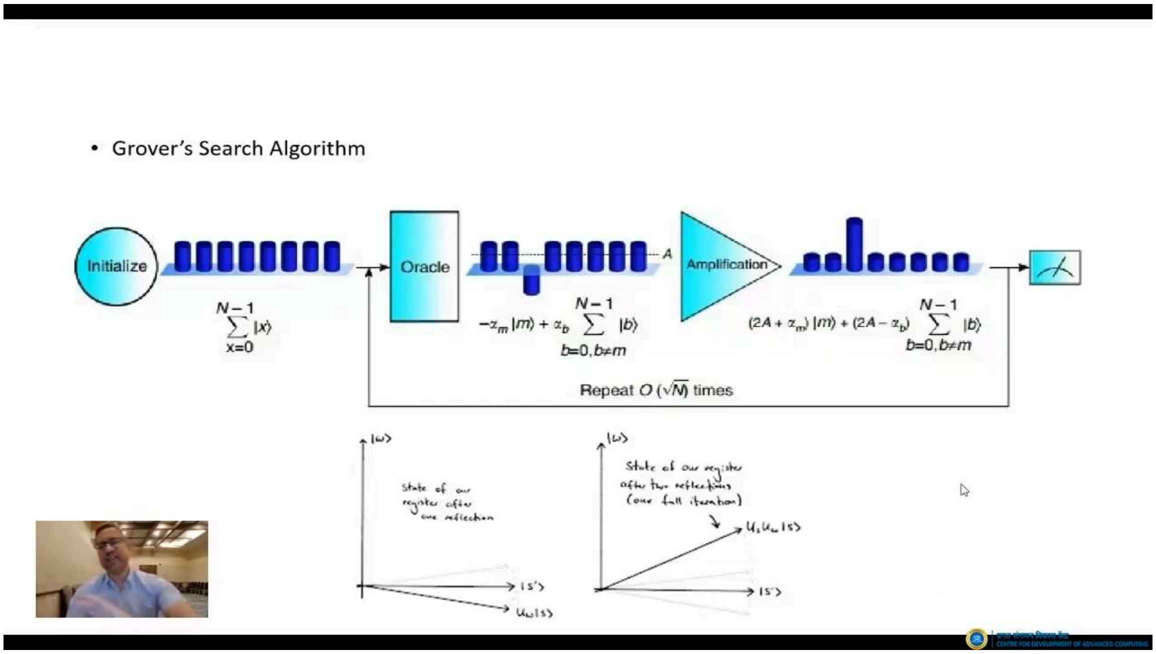
Now, suppose you have a quantum state $ \vert \psi \rangle $ that you want to copy. If you could clone this state perfectly, you would have two copies: $ \vert \psi \rangle $ and $ \vert \psi \rangle $, each in the same state as the original. However, this contradicts quantum mechanics for two reasons:

- **Measurement**: If you measure $ \vert \psi \rangle $ to copy it, the state collapses, and you get a definite outcome, say $ \vert 0 \rangle $. Now, you cannot simultaneously have $ \vert \psi \rangle $ and $ \vert 0 \rangle $ in a cloned copy, as that would require the clone to be in a superposition, which violates the measurement principle.

- **Entanglement**: Cloning would also violate the principle of entanglement. If $ \vert \psi \rangle $ was part of an entangled state with another qubit, cloning $ \vert \psi \rangle $ would also require cloning the entangled state, which is not possible without altering the original entanglement properties.

As a result, the No-Cloning Theorem fundamentally limits our ability to copy arbitrary quantum states. This theorem has significant implications for quantum computing, cryptography, and communication protocols, as it restricts certain operations that are possible in classical information processing.

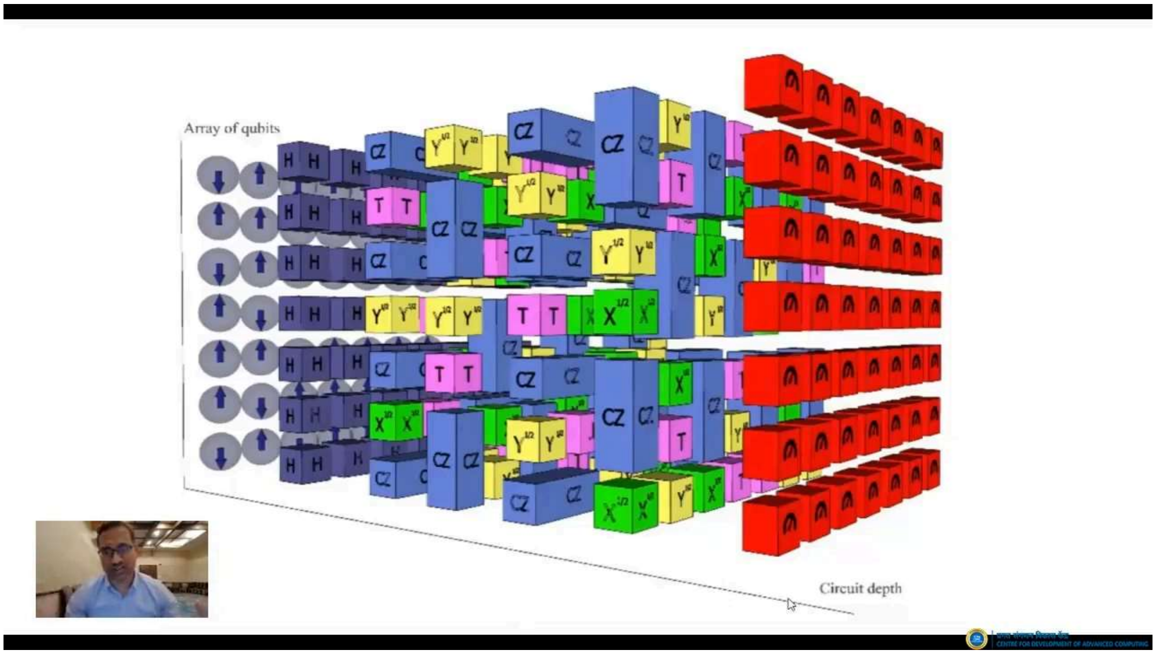




Qiskit Textbook (Algorithms) - https://github.com/Qiskit/textbook/tree/main/notebooks/ch-algorithms

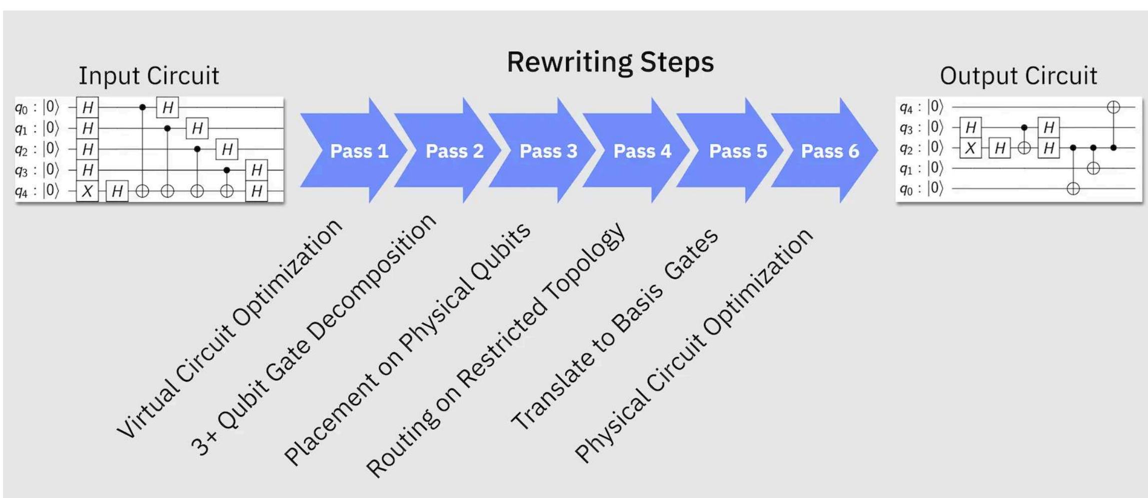
Grovers - https://github.com/SRI-International/QC-App-Oriented-Benchmarks/tree/master/grovers

Quantum Circuit in larger scale



Space Time Volume of a Quantum Circuit Computation - The computational cost for quantum simulation increases with the volume of the quantum circuit, and in general grows exponentially with the number of qubits and the circuit depth. For asymmetric grids of qubits, the computational space-time volume grows slower with depth than for symmetric grids, and can result in circuits exponentially easier to simulate.

Transpilation - https://docs.quantum.ibm.com/api/qiskit/transpiler



1. **Virtual Circuit Optimization:** This involves optimizing the abstract representation of the quantum circuit before mapping it to the physical hardware. It may include simplifications and optimizations to reduce the number of gates or improve gate sequences.

2. **3-Qubit Gate Decomposition:** Many quantum computers only support 2-qubit gates as their native gates. Decomposition involves breaking down higher-order gates, such

as 3-qubit gates, into combinations of native 2-qubit gates.

3. **Placement on Physical Qubits:** Assigning qubits from the abstract circuit to physical qubits on the quantum hardware. This step aims to minimize errors and optimize connectivity.

4. **Routing on Restricted Topology:** Quantum computers often have restricted connectivity between qubits due to physical limitations. Routing involves finding paths through the available qubits to execute the circuit, considering the connectivity constraints.

5. **Translate to Basis Gates:** Expressing the circuit in terms of the native gate set supported by the quantum hardware. This often includes single-qubit gates (like X, Y, Z rotations) and a set of 2-qubit gates.

6. **Physical Circuit Optimization:** Further optimization of the circuit to minimize errors, improve gate fidelity, and maximize performance on the specific hardware.

These steps collectively ensure that a quantum algorithm, specified at a higher level of abstraction, can be executed efficiently and accurately on a real quantum processor.

Decoherence

Decoherence in quantum computing refers to the loss or destruction of quantum coherence in a quantum system. Quantum coherence is a fundamental property that allows quantum systems to exist in superpositions of states and exhibit interference phenomena, which are crucial for quantum computation.

Here are the key points about decoherence in quantum computing:

1. **Coherence and Superposition**: In quantum mechanics, particles such as qubits can exist in superpositions of states. For example, a qubit can be in a state like $\frac{1}{\sqrt{2}} (\vert 0 \rangle + \vert 1 \rangle)$, representing both 0 and 1 simultaneously with certain probabilities. This superposition is a form of quantum coherence.

2. **Interaction with Environment**: Quantum systems can lose coherence when they interact with their surrounding environment. This interaction can be through various factors such as stray electromagnetic fields, thermal fluctuations, and interactions with nearby particles. These environmental influences can cause the quantum system's state to become entangled with the environment, leading to decoherence.

3. **Loss of Quantum Properties**: Decoherence results in the loss of quantum properties like superposition and entanglement. As coherence diminishes, the quantum system behaves more like a classical system, where states are well-defined and do not exhibit quantum effects.

4. **Impact on Quantum Computation**: Decoherence is a significant challenge in quantum computing. Quantum algorithms rely on maintaining coherence to perform calculations efficiently, leveraging properties like quantum parallelism and interference. Decoherence introduces errors and makes it difficult to sustain the delicate quantum states required for computations.

5. **Error Correction and Mitigation**: To address decoherence in quantum computing, techniques like quantum error correction and error mitigation are employed. Quantum error correction codes redundantly encode quantum information to protect against decoherence-induced errors. Error mitigation techniques aim to reduce the impact of errors caused by decoherence without requiring full error correction.

In summary, decoherence in quantum computing is the process by which quantum systems lose their coherence due to interactions with the environment, leading to the degradation of quantum properties and impacting the reliability and efficiency of quantum computations.

In [ ]: