**QUANTUM BASICS**

By: Basa Gnana Chandrika

**Abstract**

This report explores the foundational principles and operational mechanics of quantum computing, with a focus on qubit behaviour, quantum gates, and circuit design. Core quantum phenomena such as superposition, entanglement, and measurement theory are examined to build a strong conceptual base. A detailed analysis of quantum gate operations—including Pauli matrices, Hadamard, rotation, and controlled gates—demonstrates how quantum information is manipulated and measured. The project also investigates the mathematical underpinnings of quantum states using Dirac notation, matrix representations, and probability amplitudes. Emphasis is placed on contrasting classical and quantum systems to highlight the computational power of qubits.

While practical implementation challenges like qubit decoherence and error correction remain, this foundational study provides a crucial stepping stone for understanding more advanced quantum algorithms and their future applications in fields such as cryptography, optimization, and machine learning. The work also incorporates simulations and visualizations to validate theoretical insights and demonstrate gate functionalities. Overall, this report contributes to the growing understanding of quantum mechanics in computational frameworks and sets the stage for the integration of quantum algorithms into real-world problem-solving scenarios.

**Keywords:** Quantum, Qubits, Quantum Gates, IBM, Superposition, Entanglement

**Content**

[Content 2](#_Toc198920673)

[List of Abbreviations 6](#_Toc198920674)

[1 Introduction to Quantum Computing 7](#_Toc198920675)

[2 Classical vs Quantum System 8](#_Toc198920676)

[3 Qubits and Quantum States 9](#_Toc198920677)

[3.1 Qubits 9](#_Toc198920678)

[3.2 Quantum States 9](#_Toc198920679)

[3.3 Key Differences between Classical Bits and Qubits 9](#_Toc198920680)

[3.4 Applications 9](#_Toc198920681)

[4 Quantum Theory: Core Notations and Properties 11](#_Toc198920682)

[4.1 Qubit Notation(Ket) 11](#_Toc198920683)

[4.2 Qubit Notation(Bra) 11](#_Toc198920684)

[4.3 Kets of 11](#_Toc198920685)

[4.4 Superposition 12](#_Toc198920686)

[4.5 Hermitiam Operators 12](#_Toc198920687)

[4.6 Principle of Entanglement 12](#_Toc198920688)

[4.6.1 Bell States: 13](#_Toc198920689)

[4.6.2 Bell Measurement 13](#_Toc198920690)

[4.7 Amplitudes 14](#_Toc198920691)

[4.8 Interference 14](#_Toc198920692)

[4.9 Multi-Qubit Superposition (Matrix Representation) 14](#_Toc198920693)

[4.10 Orthogonality 15](#_Toc198920694)

[4.11 Normalization 15](#_Toc198920696)

[4.12 Inner Product (Bra-Ket Notation) 15](#_Toc198920698)

[4.13 Outer Product 16](#_Toc198920699)

[4.14 Measurememt and Probability 16](#_Toc198920700)

[4.15 Quantum Teleportation & No-Cloning Theorem 17](#_Toc198920701)

[5 Quantum Gates and Circuits 18](#_Toc198920702)

[5.1 Pauli-X Gate (NOT Gate) 18](#_Toc198920703)

[5.2 Pauli-Y Gate 18](#_Toc198920704)

[5.3 Pauli-Z Gate 18](#_Toc198920705)

[**5.3.1** **Ket of**  19](#_Toc198920706)

[**5.3.2** **Ket of**  19](#_Toc198920707)

[**5.3.3** **Action on**  19](#_Toc198920708)

[5.4 Hadamard Gate (H) 19](#_Toc198920709)

[5.5 S-Gate 20](#_Toc198920710)

[5.6 T Gate (π /8 Gate) 20](#_Toc198920711)

[5.7 CNOT Gate (Controlled-NOT) 21](#_Toc198920712)

[5.8 RX Gate (Rotation around X-axis) 23](#_Toc198920713)

[5.9 RY Gate (Rotation around Y-axis) 24](#_Toc198920714)

[5.10 RZ Gate (Rotation around Z-axis) 24](#_Toc198920715)

[6 Conclusion 26](#_Toc198920716)

List of Figures

No figures found

List of Tables

No table of figures entries found.

**List of Abbreviations**

|  |  |  |  |
| --- | --- | --- | --- |
| **SA** | | Shor’s algorithm | |
| **GA** | | **Grover’s Algorithm** | |
| **QFT** | | **Quantum Fourier Transform** | |
| **QAOA** | | **Quantum Approximate Optimization Algorithm** | |
| **ML** | Machine Learning | |
| **VQE** | Variational Quantum Eigen solver | |
| **H** | | Hadamard Gate | |  |  |
| **QML** | | Quantum Machine Learning | |  |  |
| **VQA** | | Variational quantum algorithms | |  |  |
| **RZ** | | Phase Rotation Gate | |  |  |
| **\*** | | Control Gate Indicator | |  |  |
| ⊕ | | Controlled NOT gate | |  |  |
| **QEC** | | Quantum Error Correction | |  |  |
|  | |  | |  |  |
|  | |  | |  |  |
|  | |  | |  |  |
|  | |  | |  |  |
|  | |  | |  |  |

# Introduction to Quantum Computing

Quantum computing is a new area of computer science that is based on the laws of quantum physics—the same laws that explain how very small particles like electrons and photons behave. It is different from classical computing in both how information is stored and how it is processed.

In a classical computer, information is stored in bits, which can be either a 0 or a 1. These bits are used to perform calculations, run software, and carry out all the operations we associate with computing. However, some problems—like breaking very large encryption codes, simulating how molecules behave in chemistry, or finding the best route for deliveries across a city—are extremely hard and time-consuming for even the fastest classical computers.

Quantum computers use quantum bits, or qubits, which are fundamentally different from classical bits. Thanks to a property called superposition, a qubit can be in a state of 0, 1, or both at the same time. This means a quantum computer can work on many possible solutions at once. Another important property is entanglement, where qubits become linked so that the state of one qubit depends on the state of another, even across distance. These quantum features allow certain problems to be solved much faster than with classical computing.

The idea of quantum computing was introduced in the 1980s by scientists such as Richard Feynman and David Deutsch, who realized that classical computers were not efficient at simulating the behaviour of quantum systems. This led to the development of quantum algorithms—special sets of instructions that quantum computers can follow. Two famous examples are Shor’s algorithm, which can break large numbers into prime factors very quickly (important in cryptography), and Grover’s algorithm, which speeds up searching through large databases.

While today’s quantum computers are still in the early stages and not yet ready for widespread use, research is rapidly progressing. Companies like IBM, Google, and several startups are building and testing quantum hardware. In the future, quantum computing is expected to transform industries such as cybersecurity, medicine, finance, and artificial intelligence.

In summary, quantum computing represents a major shift in how we think about computation. By using the strange and powerful rules of quantum physics, it has the potential to solve problems that are beyond the reach of classical computers.

# Classical vs Quantum System

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Classical Systems** | **Quantum Systems** |
| **Basic Principles** | Deterministic; outcomes can be precisely predicted. | Probabilistic; outcomes are uncertain until measured. |
| **State Representation** | Defined by measurable values (e.g., position, speed). | Described by probabilities and wavefunctions. |
| **Determinism** | Predictable and follows clear cause-and-effect. | Uncertainty in predicting exact outcomes. |
| **Superposition** | Only one state at a time. | Can be in multiple states at once. |
| **Entanglement** | No particle connection over distance. | Particles can be entangled, affecting each other across distances. |
| **Information Processing** | Classical bits (0 or 1). | Quantum bits (qubits), can be 0, 1, or both. |
| **Measurement** | Does not affect the system. | Measurement forces the system to a definite state. |
| **Applications** | Used in everyday technology (e.g., cars, machines). | Basis for quantum computing, cryptography, and lasers. |

# Qubits and Quantum States

## Qubits

* Definition: The basic unit of quantum information, similar to a classical bit, but it can be in a superposition of both 0 and 1 at the same time.
* **Superposition**: A qubit can exist in multiple states at once, unlike classical bits which are either 0 or 1.
* **State Representation**: A qubit's state is written as ∣ψ⟩=α∣0⟩+β∣1⟩, where α and β are probabilities.
* **Measurement**: Upon measurement, a qubit collapses to either 0 or 1 based on the probabilities of α and β.

## Quantum States

* **Definition**: A quantum state describes the condition of a quantum system. It can be pure (exact state) or mixed (uncertain state).
* **Pure States**: Fully known states that can be represented as a combination of basis states.
* **Mixed States**: Statistical mixtures of pure states, represented by a density matrix.
* **Superposition**: Quantum states can exist in a combination of multiple possibilities at the same time.
* **Entanglement**: When two or more quantum systems are linked, the state of one system affects the others, even at a distance.

## Key Differences between Classical Bits and Qubits

* **Classical Bits**: Can only be 0 or 1.
* **Qubits**: Can be in a superposition of both 0 and 1 at the same time.
* **Measurement**: Classical bits stay the same after measurement, while qubits collapse into one state when measured.

## Applications

* Quantum computing uses qubits to perform complex calculations faster than classical computers.
* Quantum algorithms (like Grover's and Shor's) rely on superposition and entanglement to solve problems more efficiently.

# Quantum Theory: Core Notations and Properties

## Qubit Notation(Ket)

Denoted as:

A ket represents a column vector in a complex vector space (Hilbert space). It is used to describe a quantum state.

Example: , |1 =

A general state:

## Qubit Notation(Bra)

Denoted as:

A bra is the Hermitian conjugate (complex conjugate transpose) of a ket. It’s a row vector.

Example: ,

If : then, = |

## Kets of

In quantum computing, a qubit is represented as a vector in a two-dimensional complex Hilbert space. The standard basis vectors are: : , |1 =

A general qubit state can be written as a linear combination (superposition):

Two special states involving imaginary coefficients are:

These lie on the equator of the Bloch sphere and are orthogonal to each other. They're important in understanding phase, complex amplitudes, and interference in quantum systems.

## Superposition

Superposition means that a quantum bit (qubit) can be in a combination of both 0 and 1 at the same time, unlike a classical bit which is either 0 or 1.

Mathematically, a qubit in superposition is written as:

, where α and β are complex numbers.

## Hermitiam Operators

In quantum mechanics, observables (like energy, momentum, spin) are represented by Hermitian operators H — meaning:

This guarantees:

* Eigenvalues are real → which makes sense for physical measurements.
* Eigenstates form a complete orthonormal basis → any quantum state can be decomposed in terms of these.

**Eigen Value Equation:**

Consider: (Hermitian Operator)

Two-Level System:

For the above 2 operations, Eigen Value equations are:

* is an eigen state of with corresponding eigen value .
* is an eigen state of with corresponding eigen value .

**Unitary Operations:** Quantum state evolution (without measurement) is governed by unitary operators U which preserve norm:

## Principle of Entanglement

When two or more qubits are entangled, the state of one qubit depends on the state of another, even if they are far apart.

Measuring one qubit instantly affects the other — this leads to strong correlations not possible in classical systems.

Example: Entangled state (Bell state):

If you measure the first qubit and get 0, the second is definitely 0.

If you get 1, the second is definitely 1.

### Bell States:

There are four maximally entangled 2-qubit states (Bell states):

### Bell Measurement

A Bell measurement is a measurement in the Bell basis. Instead of measuring in-terms of , , , , we measure the system in-terms of Bell States:

Given , find i and j

Example:

For the above amplitude: we apply CNOT, then we get

CNOT:

After applying Hadamard Gate:

|  |  |
| --- | --- |
| **Input (Bell State) -** | **Output (ij)** |
|  | 00 |
|  | 01 |
|  | 10 |
|  | 11 |

## Amplitudes

The probability amplitude α\alphaα and β\betaβ describe the probability of measuring the qubit in the respective basis states. The probability is given by the squared modulus of the amplitude.

Example: if and , then:

,

The system has equal probability to be measured in or .

## Interference

Interference in quantum mechanics arises because probability amplitudes (complex numbers) can add or subtract due to their phase.

* Constructive Interference: Occurs when amplitudes reinforce each other → increases the probability of an outcome.
* Destructive Interference: Occurs when amplitudes cancel → reduces or eliminates the probability of certain outcomes.

Quantum algorithms exploit interference:

* **Grover’s algorithm**: Uses constructive interference to boost the right answer.
* **Quantum Fourier Transform (QFT)**: Key to Shor’s algorithm, based on interference patterns.

## Multi-Qubit Superposition (Matrix Representation)

**Two-Qubit Superposition**: A two-qubit system can be described as a linear combination of the four possible basis states: , , , :

Example: Consider the state:

In matrix form:

## Orthogonality

## Orthogonal States: Two quantum states are orthogonal if their inner product is zero:

Example: Consider the states and |1 = . Their inner product is:

Hence, and are orthogonal.

## Normalization

## A quantum state must be normalized, meaning the sum of the squared magnitudes of the amplitudes must equal 1:

**Example**: For the qubit state, + , we have:

and

The normalization condition is satisfied:

## Inner Product (Bra-Ket Notation)

The inner product of two quantum states is a scalar that represents the "overlap" of the states.

Example: Let . Then the inner product is:

This shows that the states are orthogonal.

## Outer Product

The outer product of two quantum states and results in a matrix, representing a projection onto the state .

Example: Let The outer product is:

## Measurememt and Probability

When a qubit is measured, it collapses to one of the possible basis/eigen states. The probability of measuring a specific state is the squared magnitude of the amplitude.

Example: Cosnider the state , :

Therefore, the probability of measuring is 50% each.

*„When we measure an observable of a quantum system, the outcome is random, but the probability of each outcome is determined by the system’s state. The exact result of any one measurement cannot be predicted, only the probabilities.“*

Suppose AAA is an observable with eigenstates {}and a quantum system is in state , then:

**Born Rule**: The probability of measuring the outcome​ is:

**Post-Measurement State:** If outcome is obtained, the state collapses to the corresponding eigen state .

## Quantum Teleportation & No-Cloning Theorem

**Quantum Teleportation:** Communicating information over arbitrary long distances using the power of quantum entanglement.

Quantum teleportation allows a qubit’s state to be transmitted from Alice to Bob without physically sending the qubit.

Steps:

* Alice and Bob share an entangled pair.
* Alice entangles her qubit with an unknown state and performs a Bell measurement.
* Alice sends 2 classical bits to Bob (her measurement result).
* Bob applies a unitary correction (like Pauli-X or Z) to reconstruct the original qubit.

The actual state is not copied or moved, but recreated at a distance, relying on quantum entanglement and classical communication.

**No-Cloning Theorem:** States that it is impossible to create an identical copy of an arbitrary unknown quantum state.

Reason: Quantum State can’t be copied because any attempt to clone them would require a universal operation that preserves inner product, which is mathematically impossible for unknown state.

# Quantum Gates and Circuits

Quantum gates are the building blocks of quantum circuits. Unlike classical logic gates, quantum gates manipulate qubits using unitary operations and can place them into superposition, introduce phase shifts, or entangle them with other qubits.

## Pauli-X Gate (NOT Gate)

Action : Flips the qubit state ; equivalent to a classical NOT gate.

Matrix Representation:

Action on Pure States :

Performs a π rotation about the X-axis on the Bloch sphere.

## Pauli-Y Gate

Action : Combines Bit-Flip and Phase-Shift ; introduces complex amplitudes.

Matrix Representation :

Action on Pure States :

Rotation of π radians about the Y-axis.

## Pauli-Z Gate

Action : Applies a phase flip ; leaves unchanged and flips the sign of

Matrix Representation:

Action on pure states :

### **Ket of**

The is a superposition of the computational basis states and defined as:

This state is often called the plus state, and it has the following properties:

Representation in vector form:

On the Bloch sphere: It lies on the x-axis (along θ=π/2, ϕ=0).

### **Ket of**

The is a superposition of the computational basis states and defined as:

This state is often called the plus state, and it has the following properties:

Representation in vector form:

On the Bloch sphere: It lies on the x-axis (along θ=π/2, ϕ= π).

### **Action on**

Above actions represents Phase-Flip.

## Hadamard Gate (H)

Action: It transforms a qubit from a definite state (either |0⟩ or |1⟩) into a superposition state, where it has an equal probability of being measured as |0⟩ or |1⟩. It effectively creates a balanced state of both possibilities.

Matrix Representation:

Action on Pure States:

Superposition: Hadamard gate's primary function is to create a superposition, meaning the qubit exists in a state that is a combination of both |0⟩ and |1⟩ simultaneously.

## S-Gate

Action: Flipping the bit by 90 degrees (or) Adds a phase of π/2 ​ to the ∣1⟩ state.

Matrix Representation:

Action on Pure States:

Used in phase manipulation and error correction.

## T Gate (π /8 Gate)

Action: Flipping the bit by 45 degrees (or) Adds a phase of π/4 ​ to the ∣1⟩ state.

Matrix Representation:

Action on Pure States:

Essential for achieving universality in quantum circuits.

## CNOT Gate (Controlled-NOT)

The CNOT gate is a mulit-qubit gate that consists of two qubits. The first qubit is known as the control qubit and the second is known as the target qubit. If the control qubit is |1〉then it will flip the targets qubit state from |0〉to |1〉or vice versa.

Outputs:

Control qubit (qc) – decides if the action happens

Target qubit (qt) – gets flipped if control is |1〉

Matrix Representation:

The CNOT gate is a **4×4 matrix** (since it's a 2-qubit gate)

It acts on a **2-qubit state vector**:

CASE I:

Qbit goes through CNOT gate, such that the

conditional Qbit = |0>

input Qbit = |0>

Since the conditional Qbit is |0>, the NOT operation is not performed for the input Qbit. So the result is |0>.

CASE II:

Qbit goes through CNOT gate, such that the

conditional Qbit = |0>

input Qbit = |1>

Since the conditional Qbit is |1>, the NOT operation is performed for the input Qbit. So the result is |1>.

CASE III:

Qbit goes through CNOT gate, such that the

conditional Qbit = |1>

input Qbit = |0>

Since the conditional Qbit is |1>, the NOT operation is performed for the input Qbit. So the result is |1>.

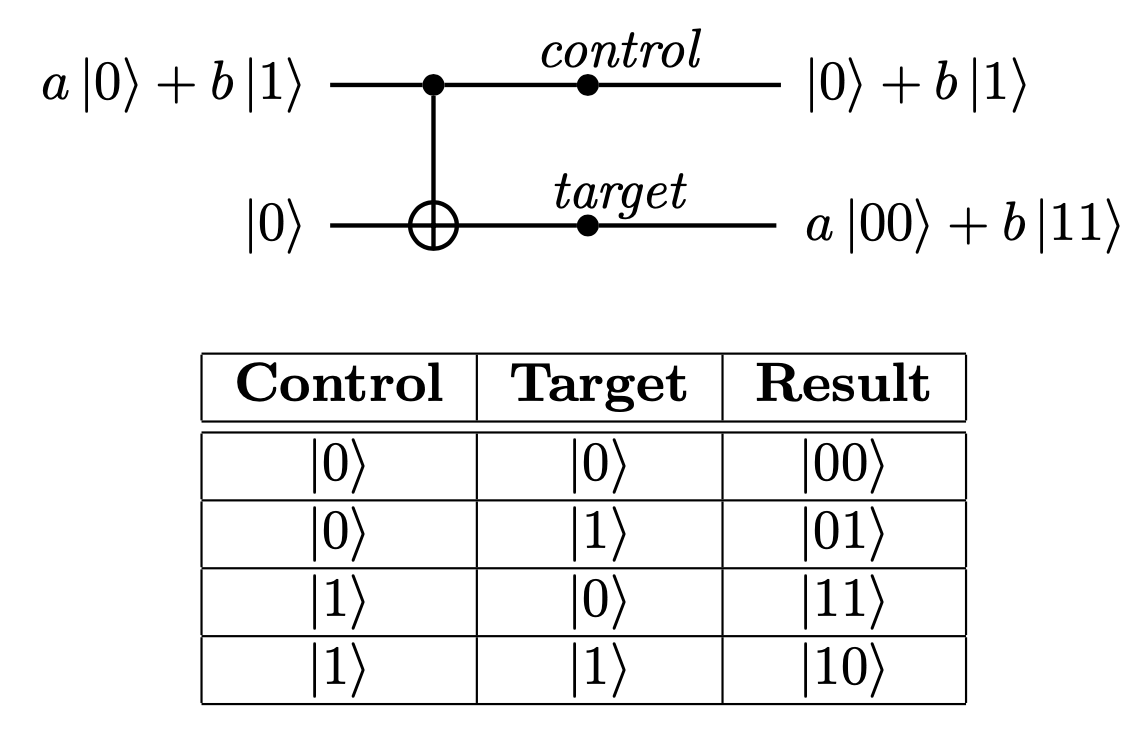
CASE IV:

Qbit goes through CNOT gate, such that the

conditional Qbit = |1>

input Qbit = |1>

Since the conditional Qbit is |1>, the NOT operation is performed for the input Qbit. So the result is |0>.



## RX Gate (Rotation around X-axis)

Same phase – Bit flip

Action: Rotates the qubit state **around the X-axis** of the Bloch sphere by an angle θ.

Matrix Representation:

Since

Action on Pure States:

For

For

If , then:

## RY Gate (Rotation around Y-axis)

Same phase – Bit flip

Action: Rotates the qubit state **around the Y-axis** of the Bloch sphere by an angle θ.

Matrix Representation:

Since

Action on Pure States:

For

For

If , then:

## RZ Gate (Rotation around Z-axis)

Same phase – Bit flip

Action: Rotates the qubit state **around the Z-axis** of the Bloch sphere by an angle θ.

Matrix Representation:

Since

Action on Pure States:

For

For

If , then:

# Conclusion