Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Exploring Discrete Algorithms in Quantum Computing

Sulochana Sonkamble

sbsonkamble@pict.edu

Pune Institute of Computer Technology

Balwant Sonkamble

Pune Institute of Computer Technology

Research Article

Keywords: Quantum Computing, Quantum Cryptography, Superposition, Entanglement, Discrete Quantum Algorithms, Quantum AI, Quantum Machine Learning

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11232251/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Exploring Discrete Algorithms in Quantum Computing

SulochanaSonkamble[1],Balwant Sonkamble[2]

[1]*Department of Information Technology, Pune Institute of Computer Technology, India*

[1]Email:[1]sbsonkamble@pict.edu

[1]ORCID ID: 0000-0002-6081-8838

[2]*Department of Computer Engineering,* Pune Institute of Computer Technology*, India*

[2]*Email:* [2]basonkamble@pict.edu

[2]ORCID ID:0009-0001-8681-6347

ABSTRACT

Quantum computing introduces a new computational paradigm by extending the classical binary representation of information through quantum bits, which can exist in superposed states and exhibit entanglement. This work presents a systematic study of fundamental concepts, computational mechanisms, and discrete algorithms in quantum computing. The introduction discusses binary bits and qubits, classical logical gates, and their quantum counterparts, highlighting the reversible nature of quantum computation. The materials and methods examine XOR-based classical cryptography, quantum cryptography, quantum teleportation, and mathematical representations of quantum states using probability amplitudes, superposition, and entanglement. The results and discussion focus on the quantum computing model and the implementation of reversible logic through important quantum gates, particularly the Toffoli and Fredkin gates. The role of superposition and entanglement in manipulating quantum information and performing computations is also analyzed. Based on these foundations, the proposed framework emphasizes discrete quantum algorithms for solving problems involving Boolean functions, binary strings, search spaces, integers, graphs, and combinatorial structures. Representative quantum algorithms, including Deutsch, Deutsch–Jozsa, Bernstein–Vazirani, Simon, Grover, and Shor, demonstrate the application of quantum principles to discrete computational problems. The study further explores the relationship between quantum computing, Artificial Intelligence (AI), and Quantum Machine Learning (QML), where quantum representations and algorithms can support optimization, classification, search, and data-processing tasks. The work concludes by identifying future research directions involving quantum AI, hybrid quantum–classical architectures, error-resilient quantum systems, scalable hardware, and practical discrete quantum algorithms.

KEYWORDS

Quantum Computing, Quantum Cryptography, Superposition, Entanglement, Discrete Quantum Algorithms, Quantum AI, Quantum Machine Learning.

1. INTRODUCTION

Information theory and quantum mechanics have emerged as two influential foundations of modern information processing, communication, and intelligent computing. Their convergence has given rise to quantum information theory, which provide new approaches for representing, transmitting, and processing information and has motivated the development of novel quantum algorithms, communication protocols, and computational architectures. Classical information processing is primarily based on discrete binary states and Boolean logic, whereas quantum computing extends this framework by exploiting fundamental principles of quantum mechanics, including superposition, entanglement, interference, and measurement [1]. These properties provide new computational mechanisms for addressing selected problems that may be difficult to solve efficiently using conventional computing systems. However, the computational advantage of quantum computing depends on the mathematical structure of the problem, the availability of suitable algorithms, and the capabilities of the underlying quantum hardware.

Classical computers represent information using bits, with each bit having a definite value of either 0 or 1. In contrast, a quantum bit (qubit) can exist as a linear combination of the computational basis states 0 and 1 [2]. This representation allows quantum systems to manipulate probability amplitudes and exploit interference during the computation. When multiple qubits are combined, entanglement can establish correlations between quantum states that have no direct classical equivalent. These properties form the foundation for quantum algorithms and enable computational approaches that differ fundamentally from conventional binary processing.

The logical foundation of computation also changes when classical systems are extended to quantum systems. Classical computers employ logical gates such as AND, OR, NOT, and XOR to manipulate binary information. Quantum computation, however, requires reversible and unitary operations because quantum evolution must preserve the normalization of quantum states. Quantum gates such as Pauli-X, Pauli-Z, Hadamard, CNOT, Toffoli, and Fredkin therefore provide fundamental mechanisms for constructing quantum circuits. The Hadamard gate is particularly important for generating superposition, while the CNOT gate can create entanglement between qubits. The Toffoli gate performs a controlled-controlled-NOT operation, whereas the Fredkin gate provides controlled swapping of quantum states. These reversible gates establish an important connection between classical discrete logic and quantum information processing and provide the computational building blocks for implementing quantum algorithms.

Quantum information processing also introduces new approaches to cryptography and communication [3]. XOR is a fundamental operation in classical cryptographic mechanisms because of

its simple and reversible logical properties. Quantum cryptography, in contrast, utilizes quantum states, measurement, and fundamental physical principles to establish secure communication mechanisms. Quantum teleportation demonstrates the use of entanglement for transferring an unknown quantum state between distant locations using a shared entangled state and classical communication, without physically transmitting the original quantum state. These processes can be mathematically described using state vectors, probability amplitudes, quantum operators, and measurement principles. Consequently, superposition and entanglement are important not only for quantum computation but also for quantum communication and information security.

Quantum optics provides an important physical platform for demonstrating quantum information concepts. Photons, or particles of light, can encode quantum information through properties such as polarization. Horizontal and vertical polarization can be associated with computational basis states, while intermediate polarization orientations can represent quantum superposition [4]. For example, polarization angles of 0^0 and 90^0 are commonly associated with horizontal and vertical polarization, respectively. The ability to control photon polarization has made optical systems valuable experimental platforms for investigating quantum phenomena such as quantum teleportation and quantum key distribution. Photon-based representations therefore provide an intuitive physical interpretation of qubits, superposition, measurement, and quantum communication.

A central focus of this study is the analysis and organization of discrete algorithms in quantum computing. Discrete algorithms operate on finite or countable structures such as Boolean functions, binary strings, integers, graphs, search spaces, and combinatorial problems. Quantum computation provides alternative mechanisms for addressing these problems through quantum-state preparation, quantum gates, oracles, interference, amplitude amplification, and measurement. Deutsch and Deutsch–Jozsa algorithms demonstrate how quantum interference can be used to analyze Boolean functions. The Bernstein–Vazirani algorithm identifies hidden binary information, while Simon's algorithm demonstrates the extraction of hidden structures from discrete functions. Grover's algorithm applies amplitude amplification to unstructured search and provides a quadratic improvement in query complexity under its standard oracle model. Shor's algorithm employs quantum period finding and the Quantum Fourier Transform to address integer factorization in the standard idealized quantum computational model. These algorithms demonstrate that quantum computational advantage results from exploiting problem structure through superposition, interference, and other quantum operations rather than merely evaluating all possible solutions simultaneously.

The integration of quantum computing with Artificial Intelligence (AI) and Quantum Machine Learning (QML) further expands the potential applications of discrete quantum algorithms. Quantum AI can investigate quantum approaches for intelligent search, optimization, decision support, pattern

processing, and computational reasoning. QML combines quantum-state representations and parameterized quantum circuits with classical machine-learning techniques and may support applications involving classification, feature mapping, optimization, and combinatorial learning. Nevertheless, current quantum computing systems face significant practical challenges, including noise, decoherence, limited qubit connectivity, restricted scalability, and the requirement for effective error correction. Hybrid quantum–classical architectures therefore provide a practical approach for combining the computational capabilities of quantum processors with the mature processing and optimization capabilities of classical computers [24].

Motivated by these developments, the primary objective of this study is to establish an integrated understanding of discrete algorithms in quantum computing by connecting classical discrete logic, quantum information processing, quantum gates, quantum communication, quantum algorithms, Quantum AI, and QML. The study specifically examines the differences between bits and qubits, investigates reversible gates such as Toffoli and Fredkin, analyzes superposition and entanglement as computational resources, and considers representative discrete quantum algorithms for Boolean functions, hidden structures, search, factorization, and combinatorial optimization. The motivation arises from the increasing complexity of computational problems and the limitations of conventional computing for selected large-scale discrete problems. Although quantum computing offers new computational mechanisms, systematically transforming classical discrete problems into quantum representations and selecting appropriate quantum algorithms remains challenging.

A research gap exists because many existing studies examine individual quantum algorithms, quantum gates, quantum cryptography, quantum communication, or QML independently, while comparatively less attention is given to an integrated architecture connecting these concepts from fundamental quantum representation through algorithmic processing to intelligent applications. To address this gap, this study proposes a layered quantum–classical framework consisting of discrete problem formulation, classical-to-quantum encoding, quantum-state preparation, reversible quantum-gate processing, discrete quantum algorithm selection, quantum measurement, classical post-processing, and Quantum AI/QML integration. The novelty of the proposed framework does not rely on introducing a new quantum algorithm; instead, it provides a systematic architecture for mapping different classes of discrete problems to appropriate quantum computational mechanisms and connecting established algorithms—including Deutsch, Deutsch–Jozsa, Bernstein–Vazirani, Simon, Grover, Shor, and QAOA, with quantum information processing and intelligent computing. The framework emphasizes superposition, entanglement, interference, amplitude amplification, and the Quantum Fourier Transform as key mechanisms underlying discrete quantum algorithms. Its hybrid design also considers the practical limitations of current noisy quantum hardware and establishes a foundation for future research

in discrete optimization, cryptography, Quantum AI, and Quantum Machine Learning. Overall, the proposed work provides an integrated perspective for understanding, organizing, and extending discrete algorithms within the evolving field of quantum computing.

1.1 BINARY BITS AND QUBITS

There exists some kind of well defined logic for every computation. The classical computation is being performed using Boolean logic which uses variables 0 and 1 values. The physical interpretation of 0 and 1 is the voltage On/Off. But in case of quantum computation, quantum mechanics provides a new set of rules that go beyond this classical paradigm. The basic variable in quantum computing is a quantum bit which is represented as a vector in a two dimensional complex Hilbert space. The logic that can be implemented with such qubits is quite distinct from classical Boolean logic and this is what has made Quantum computing exciting by opening new possibilities. In recent years quantum computing becomes a forefront research in the field of algorithms, cryptography and artificial intelligence [1].We can describe qubits as mathematical objects with certain specific properties. The beauty of treating qubits as mathematical objects is that it gives us the freedom to construct a general theory of quantum computation, which does not depend upon a specific system for its realization. The task of information processing can be performed using quantum mechanical principles. And when quantum principles are applied on information processing it gives the concept of quantum computing. In quantum mechanics the state of a physical system is represented by its wave function which contains all information to describe the state of the system completely. Contrary to classical computing we carry out computations using quantum states which follow properties of Quantum mechanics. Changes occurring to a quantum state can be explained using the language of quantum computation. As classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry out and manipulate the quantum information. This paper addresses the basic properties of classical and quantum gates and a comparison is made between them with emphasising the shortcomings of classical gates [20].

1.2 LOGICAL GATES IN CLASSICAL COMPUTER

In 1936, Scientists, Church and Turing has been introduced a classical computation theory and presented their investigation into the characteristics of computability . Logic gates and logic circuits played a major role in theory of computation to develop the optimal structure of classical logic circuits [20]. A gate is said to be logically reversible if we can uniquely determine the input values from the output values otherwise the gate is said to be logically irreversible. Table (1) shows the input - output of the logical gates in classical computer. The notions of boolean functions of classical logic gates: AND,

OR, XOR, NAND, NOR are given below in quations (1),, (2), (3), (4) and (5) respectively.

AND-GATE:

It is defined as a Boolean Function:

$$f(x,y) = \begin{cases} 1, & if\ x = y = 1 \\ 0, & otherwise \end{cases} \quad (1\text{-a})$$

The result can be written as:

$$f(x,y) = xy \qquad (Meaning: product\ of\ x\ and\ y)$$

(1-b)

OR - GATE:

It is defined as a Boolean Function:

$$f(x,y) = \begin{cases} 0, & if\ x = y = 1 \\ 1, & otherwise \end{cases} \quad (2\text{-a})$$

The result can be written as:

$$f(x,y) = x + y \qquad (Meaning: plus\ of\ x\ and\ y) \quad (2\text{-b})$$

XOR - GATE:

XOR (Exclusive – OR) is a variant of OR function

It is defined as a Boolean Function:

$$f(x,y) = \begin{cases} 0, & if\ x = y = 0\ or\ x = y = 1 \\ 1, & otherwise \end{cases} \quad (3\text{-a})$$

The result can be written as:

$$f(x,y) = x \oplus y \qquad (Meaning: plus\ of\ x\ and\ y) \quad (3\text{-b})$$

The Exclusive-OR is an interesting gate, when both inputs are 1, the output is zero. In exclusive-OR, interpreted as exclusive-ORing one boolean variable onto another, "exclusive OR B onto A" means the same as A = A XOR B. In other words, A is changed to the result of A XOR B and this is done in a single atomic operation. The value of B is XOR-ed onto the value in A, and the result remains in A.

NAND-GATE:

NAND (NOT-AND) is a universal gate from which any Boolean function can be derived. Let $f(x)$be a NOT function and $g(x,y)$ be a AND function. The NAND function can be defined as,

$$h = f(g) = \overline{xy} \quad (4)$$

NOR – GATE:

Similarly NOR gate is another universal gate from which any Boolean function can be derived. It can be defined as a function:

$$f(x+y) = \overline{x+y} \qquad (5)$$

Table 1 (a): Input-Output of Logical Gates in Classical Computers

Sr. No.	Input		GATE Output				
	x	***y***	**AND**	**OR**	**XOR**	**NAND**	**NOR**
1.	0	0	0	0	0	1	1
2.	0	1	0	1	1	1	0
3.	1	0	0	1	1	1	0
4.	1	1	1	1	0	0	0

1.3 LOGICAL GATES IN QUANTUM COMPUTER

The quantum computers perform operations differently based on quantum mechanics laws. Its basic unit is quantum bits, called as qubits. Qubit has a quaternary nature, it exists in the states corresponding to the logical values 0 or 1 like a classical bit, and additional a superposition state. A qubit represents a bit of information that can be both zero and one simultaneously. Thus, a computer working on a qubit rather than a standard bit can make calculations using both values simultaneously. A qubyte, is made up of eight qubits and can have all values from zero to 255 simultaneously. The memory of a classical computer is a string of 0s and 1s, and it can perform calculations on only one set of numbers. The memory of a quantum computer is a quantum state that can be a superposition of different numbers. A quantum computer can do an arbitrary reversible classical computation on all the numbers simultaneously. Specially, performing a computation on many different numbers at the same time and then interfering all the results to get a single answer, makes a quantum computer much powerful than a classical one [19]. The classical signals of voltages are represented by the state 0 and 1 which store one bit information whwreas, two bit states represented by 00,01,10,11 store two bit information, it is 2^n bits. The quantum computers can process n-inputs with only one computational step after the superposition state of n-inputs. Quantum bits are called qubit which used electron states to store the information. The states of the electron are indicated by the spin-up & spin-down and represented as 0 & 1 respectively. The photons are used as a qubit to represent polarization in vertical & horizontal direction. One qubit represents superposition of 0 & 1 states which are denoted by dirac notation, $|0\rangle$ and $|1\rangle$ respectively. A

gate is said to be logically reversible if we can uniquely determine the input values from the output values otherwise the gate is said to be logically irreversible. Quantum programs contain the equations of a sequence of quantum gates on a circuit. The behavior of these quantum gates is described by matrices of complex numbers [14]. These complex numbers in the matrices affect the probabilities of qubit measurements [20].

In quantum computing architectures, an atomic XOR gate is an important component to compute A XOR B is to XOR B onto A in an atomic operation. Exclusive OR gate can be used to perform variable assignment, like A = B. First initialize A to be 0, then Exclusive OR B onto A, this is equivalent to setting A to be A XOR B. At the end of both these steps, the value of B has been assigned to A and this has accomplished to replace assignment with two steps. In first step, initialize to 0 then perform XOR operation. In second step, perform XOR B onto A operation, it can be initialized to 0 very early in a computation before the value of B is available. Finally, the end result is replacement of value of A to value of B. This method of replacing assignment with XOR is very important for quantum computations. Quantum assignment is an irreversible operation and can be performed only once, at the beginning of quantum computations . In XOR single bit operation, entire sequences of bits can be XOR-ed with each other whereas in two bit sequences it results in another bit sequence of the same length. Following example table 1 shows the output bit sequence is obtained by XOR-ing the corresponding bits of the input.

Table1 (b): XOR Bit Sequences

Input Sequence x	1	0	1	1	0	1	1	1	0	0	0
Input Sequence y	0	1	1	0	1	1	0	1	1	0	1
Output Sequence $x \oplus y$	1	1	0	1	1	0	1	0	1	0	1

1. MATERIALS AND METHODS

1.1 CRYPTOGRAPHY WITH XOR GATE

In real world day-to-day life, it became very essential, to keep private data secure and to protect sensitive information on internet. In view of this, cryptography is used for interaction between sender and receiver, sending secret messages over a insecure channel and the message to be remain a secret. The sender of a secret message can encrypt the message whereas the receiver can decrypt the message [15]. Following table-2 shows, the encription and decryption using XOR gate through a classical communication channel. The XOR operation is used for cryptography to perform two tasks, one is

sender encrypts the message called as encryption, which convert the protected message into a sequence of bits. Other task is receiver decrypts the message called as decryption, to get a random sequence of bits that only the sender and the receiver know. This random sequence is a shared secret which is a bit sequence that only know the receiver and sender and operate using the bit sequence on the same message. Finally, the encryption operation is performed on bit sequence using XOR to encrypt the message.

Table 2: Cryptography with XOR

	Encryption										
Input	1	0	0	1	0	1	1	0	1	0	Message encoded as bits
	0	1	0	1	1	0	1	1	0	1	Random sequence of bits
XOR	1	1	0	0	1	1	0	1	1	1	Encrypted message
	Decryption										
Input	1	1	0	0	1	1	0	1	1	1	Received bits, encrypted data
	0	1	0	1	1	0	1	1	0	1	Random sequence of bits
XOR	1	0	0	1	0	1	1	0	1	0	Original message, Decrypted

The encryption is performed by XORing a random bit sequence onto the message. The decryption is performed by XORing the same bit sequence onto the encrypted data. The result of the second XOR is the original message. This is represented by following equation (5), where x is the message encoded as bits, r is the random sequence .

$$x \oplus r \oplus x \oplus r \oplus r = x \tag{5}$$

1.2 QUANTUM CRYPTOGRAPHY

1.3

Quantum cryptography allows us to create secure communication protocols that are unbreakable. The cryptography protocol is called BB84, this protocol is based on the quantum properties of light. Light is a stream of photons and a photon behaves like a particle. The light is an electromagnetic wave means the electric and magnetic oscillations are at right-angles along the X-Y plane to the direction of propagation of the wave. Each photon is directed along the z-axis has its own angle, called the angle of polarization, it can be vertical or horizontal. The photon is the system, measured by the polarization angle using a filter which changes the polarization of the photon when it makes it passed the filter. The photons emitted from the light source are having all angles of polarizations as well as vertically polarized

and horizontally polarized [18].

The experiment proves that polarizing filters are not really filters but they are quantum mechanical measurement devices and quantum measurements change the state of the systems being measured. When light is polarized at the same angle as the filter, all the lights gets through deterministic behavious. When a photon's polarization is aligned with a polarizing filter, the behavior is deterministic. The photon always passes through. When a photon is anti-aligned with a polarizing filter, the behavior is deterministic. The photon is always blocked for any angle that is not aligned or anti-aligned, the behavior of the photon is probabilistic. Randomness is inherent in the behavior of quantum systems. Encryption with single sequence of bit uses the shared secrets, when any sequence of bits is XORed twice with secret sequence , the original sequence of bits is recovered. If the shared secret is a random sequence of bytes and if each shared secret is used only once, then the produced code is unbreakable. The shared secret should be used only once, if it is used for more than one message, then non-random patterns in the encoded messages will allow the code to be broken. The main objective is to securely establish a shared secret sequence of bytes through a quantum protocol.

2.3 QUANTUM TELEPORTATION

Sender and receiver are connected by two channels of communication, like internet a classical channel. Both are potentially insecure and anyone can eavesdrop on this channel. There is a quantum channel where sender can send polarized photons to the receiver. Sender first creates a long sequence of random bits and needs to send this sequence to receiver in a secure manner.

Once receiver gets these bits, both can use them as a shared secret in future communication between themselves. This sequence of random bits which sender has created will become the shared secret. The important thing is, not all of the bits which sender has created here will be in the final shared secret. A random fraction of these bits will be lost during transmission when it apply the BB84 protocol. Only a fraction of these bits will reach to receiver and become the shared secret. why some bits are corrupted on protocol.and neither sender nor receiver are able to control it during the transmission [17].

The simple approach to build an unbreakable encryption protocol is use of polarized photons through the quantum channel. Sender encodes the bits with vertical and horizontal polarization, communicates a bit value of 1 with a vertically polarized photon and encodes a bit value of 0 as a horizontally polarized photon. Using this encoding protocol, sender can send the sequence of bits to receiver. At this stage, anyone eavesdropping on the quantum channel can read the data being sent without either sender or receiver realizing that their security has been compromised. If the photon passes through the filter, then sender had sent a value 1. If the photon is blocked, then sender had sent a value 0. Next a new photon is created that encodes the same bit that sender had sent to receiver. Receiver receives

the photon sent through filter, cannot be realized that it has intercepted and resent the entire sequence of bits message that sender sent. It indicates that this is not a secure protocol means all data can read then resend it without being detected. Therefore , it is essential to establish a secure protocol at receiver. The security of the protocol is based on quantum behavior of photons of laws of physics [11].

Some of the bits will be corrupted and lost in transmission. These bits for which receiver has guessed wrong are the ones that get corrupted and are lost in transmission. To prevent the loss, sender modifies a protocol, sender chooses to encode a randomly chosen 50% of the bits in a sequence with vertical polarization for bit value 1and horizontal polarization for bit value 0. For the other 50% ,sendor encodes bit values of 0 with photon set at polarized 45^0 to the left and encodes bit values of 1 with photon set at polarized 45^0 to the right. The choice of which encoding to use for each bit is made at random. There is a 50% probability that sender will choose the vertical & horizontal encoding and there is a 50% probability that sender will choose the 45% left and right encoding.

Table 3: Encoding Data in Photon Polarization

0	1	0	0	1	1	1	0	1	0	1	1	Sender's sequence
+	+	X	+	X	X	+	X	+	X	X	X	Randomly chosen encoding scheme
-	|	\	-	/	/	|	\	|	\	/	/	Polarization angles of photons sent by sender
Quantum Channel												
X	+	+	X	X	X	+	X	X	+	X	+	Receives guess of encoding scheme
?	1	?	?	1	1	1	0	?	?	1	?	Receiver decodes photons based on his guess of encoding scheme.
	√			√	√	√	√			√		Receivers correct guess

Table (3) shows the encoding data in photon polarization angles, top row shows a bit of sequence to be send by sender, below that a random choice of encoding is given. There is one choise for each of the bit, + symbols represent the verticle and horizontal encoding scheme. When a value 0 is encoded with vertical and horizontal encoding scheme, the photon is horizontally polarized and represented by - symbol. When a value 1 is encoded with vertical and horizontal encoding scheme, the photon is vertically polarized and represented by | symbol. Other half of bits are encoded using 45^0 left and right polarization which is represented by symbol X, symbol \ indicates bit value 0 in 45^0 encoding and the photon is polarized 45^0 to the left. Symbol / indicates a bit value 1 in 45^0 encoding and the photon is polarized 45% to the right. On the receiver side, yet to know the encoding data that is used for each bit

and therefore receiver makes random guesses, 50% guesses are correct and remaining 50% guesses are wrong. Receiver decodes photons based on his guess into vertical-horizintal encoding scheme, and for these bits, it guesses 45^0 left - right encoding. In table (3), symbol √ indicates that receiver decode the photons correctly and aligns the polarizing filter according to the guess and the photon behaves in a deterministic manner which enables the receiver to decode correctly [4]. Remaining guesses are wrong and it will align the polarizing filter at a wrong angle and the behaviour of the photon is random. Randomly some photons get through the filter and some photons will not pass the filter [16].

Therefore receiver attempts to decode its bits using wrong guesses results in a random set of bits and these bits are corrupted and lost in transmission. Sender reveals the correct encoding after receiver receives the photons. If data is corrupted, sender and receiver discard the entire sequence and try again with a new set of data, eventually both are having the shared sequence code. BB84 protocol allows the sender and receiver to share a secrete key over insecure communication channel,since they have a shared secret, both can use 1-use-shared-secret encryption which is unbreakable. The security of the protocol is based on quantum behavior of photons and it is based on the laws of physics which remains secure in future. Quantum computers can crack RSA encryption using quantum properties.

2.4 MATHEMATICAL REPRESENTATION OF QUANTUM STATES

2.5

The quantum system is consists of Qubit, which is equivalent to the bit used in classical computers. The qubit has two states ON and OFF or a superposition of ON and OFF. Quantum system in superposition requires more than one numbers to define it. The quantum system can be represented mathematically using a matrix algebra known as a linear algebra and quantum states are represented by complex numbers of column matrices with dimension 4X1. The changes to this state are represented by multiplying the column matrix by a transformation matrix [5]. A quantum state without superposition is represented by a single number in the column matrix, all others are 0. Square magnitudes of the numbers in the column matrix represent probabilities.

Following figure (1) shows the matrices for quantum states , the transformation matrix of complex numbers and quantum states without superposition.

$$\begin{bmatrix} s+ti \\ u+vi \\ w+xi \\ y+zi \end{bmatrix} \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \begin{bmatrix} s+ti \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ u+vi \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ w+xi \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ y+zi \end{bmatrix}$$

Fig1: (i) Quantum states (ii) Transformation matrix of complex numbers (iii) Quantum states without superposition

When electrons move up and down in a magnetic field, this property of electrons is called spin. The spin can be used as the quantum equivalent of a bit in classical computers. When the spin is up, it is equivalent of a bit 1 and when spin is down, it is equivalent of a bit 0 . The quantum equivalent of a bit is called a qubit and this is a quantum bit. The probability of measuring spin up OR down is 1 and probability of measuring spin up AND down is 0. This property can be used as the basis for defining the states of the spin qubit represented by matrix given below in figure (2). If a qubit is in the up state, corresponding to a classical bit being 1, the matrix representation of the state will be [0 1]. If a qubit is in the down state, corresponding to a classical bit being 0, the matrix we use will be [1 0].

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

Fig2: Spin Qubit States: (i) UP state. (ii) DOWN state. Reversible state transformation:(iii) Qubit in state |0> (iv)Qubit in state |1>

In classical physics, the location-state of a particle can be specified in terms of the three axes X, Y and Z, the three coordinates are usually written as three numbers enclosed (x,y,z). Whereas in quantum physics, the numbers which describe a state are written as a column matrix, it is called a vector. A vector represents a qubit, has two elements, the upper number is the square-root of the probability of being measured in the Qubit-Off or Spin-Down state. The lower number is the square-root of the probability of being measured in the Qubit-On or Spin-Up state.

2.5 SUPERPOSITION AND ENTANGLEMENT IN QUANTUM COMPUTING

When the polarization of photon is not aligned and not anti-aligned with a measurement apparatus then photon polarization is in superposition, and the measurement on a superposed state has probabilistic behaviour. When photon polarization is aligned then qubit value is 1 value [10]. When photon polarization is anti-aligned then qubit value is 0 whereas when photon polarization is between 0 and 90 degrees, the qubit is in superposition and the result of measurement is probabilistic. In Reversible State Transformations, electron spin were used as the qubit in a quantum computer Up = 1 and Down = 0, $[1\ 0]^T$ represents the qubit is in state 0, and $[0\ 1]^T$ represents the qubit is in state 1. When, the qubits are sent as input to a NOT gate, it transforms into a reversible form. It changes bit value 1 to bit value 0 and bit value 0 to bit value 1. Applying a transformation is just matrix multiplication, that is, $[1\ 0]^T$ has been transformed to $[0\ 1]^T$ means the qubit value of 0 has become qubit value of 1. Similarly, apply the NOT operation to a qubit value 1, become the qubit value 0. In case of two qubit systems, the possible combinations are 00, 01, 10, and 11.

The state vector will have four elements, one for each combination given further steps i, ii, iii and

iv, represented in figure (3). Step (i) when both qubits are 0, the ket vector is written as |00>, step (ii) , when first qubit is 0, and the second qubit is 1, ket vector is written as |01>, step (iii) when the first qubit is 1 and the second is 0, ket vector is written as vector |10>, and step (iv) when both qubits are 1, the ket vector is written as |11>. The measurements and reversible transformations are performed on multi-qubit states in the same way of single qubit states. The superposition state can be written in ket vectors as 1 by root 2 times (|00> plus |11>). If the two qubits are 0, then the probability of the measurement is the square magnitude of number which is 0.5. Similarly if first qubit is 0, and the second qubit is 1, then probability is 0. If first qubit 1, second is 0, then probability is 0. If both qubits are 1, then the probability is square magnitude of 1 by root 2 which is 0.5.

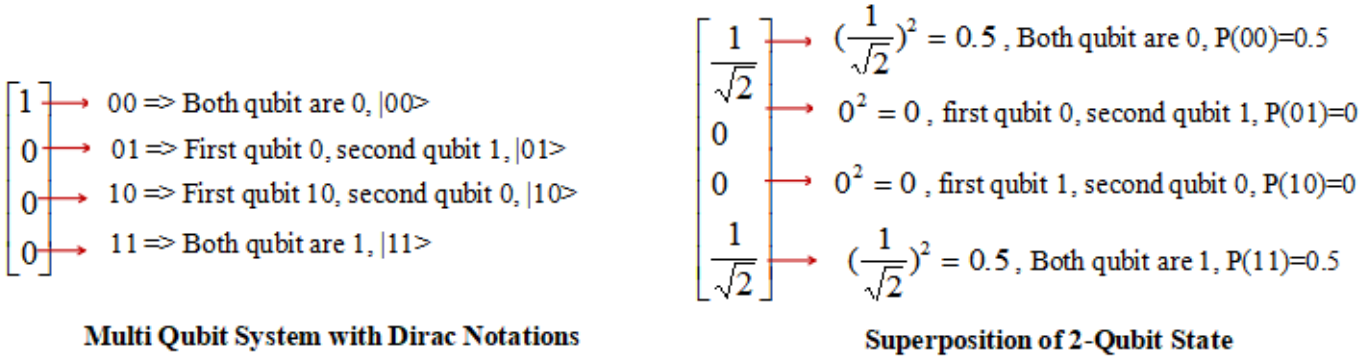


Fig3: Multi-Qubit System and Superposition States

In ket state vector of a two qubit system, the probability of both qubits being 0 is 50% whereas both qubit being 1 is 50%. Whenever, two qubit have the same value then the value of one qubit is connected to the value of the other qubit and these two qubits are entangled. Qubit entanglement behaves the same as photon entanglement. If two photons are entangled, then after measurement, they are independent of each other and the effect of entanglement is seen in the probability values. If the photon polarization is aligned, then qubit value is 1 whereas if the photon polarization is anti-aligned, then the qubit value is 0. Then qubit is entangled and qubit entanglement behaves same as photon entanglement.

3 RESULTS AND DISCUSSION

3.1 QUANTUM COMPUTING MODEL

In quantum computation, qubit state is initialized to 0, it is irreversible, the final output is obtained from the quantum computation which is irreversible and the steps in between are reversible. Figure 4 shows the quantum computations for 4-qubits input and 2-qubits output. Initial values of the bits are set on the left side indicates, the initialization is an irreversible operation and it is performed before quantum computations begin. During the quantum computations, values cannot be assigned to any of the bits as a assignment which is an irreversible operation, so it cannot be used in quantum

computations [5]. XOR gate is used onto a qubit to be initialized to 0 value. Whatever a value used with exclusive OR with a bit , it has been initialized to a value zero. Quantum computations are performed with the pattern of four bits of input and produces two bits of output as shown in figure (4). The four bits of input are initialized to whatever the input. The values cannot be assigned to the output because that would be irreversible, instead XORed values onto the output qubit. XOR is a reversible operation and can perform it during quantum computations [20]. If the output qubit are 0 at the starting, then the XOR operation is equivalent to an assignment. Both the input qubit and the output qubit are initialized in the initialization step. Two qubit are allocated at the initialization tstep to store the output and remaining 4-qubit are initialized to the input value. During the quantum computation operation, it will be XORed the output onto the bits and XOR is a reversible operation. When computation has completed, these 2-qubit are measured to produce the output.

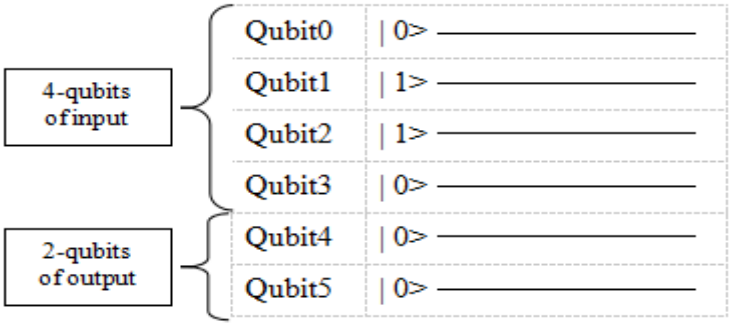


Fig4 : Quantum Computations for 4-Qubit Input and 2-Qubit Output

In quantum computations, fanout gate is used to copy the output to multiple subsequent inputs and copy qubit values 1 or 0 exactly. Most quantum computers have very few qubits as its very expensive piece of hardware, therefore it is prefered to reuse qubits during computations [6]. In quantum computation circuit model, qubit cannot be reinitialized, as it is an irreversible operation which can be done at the beginning of computation. Qubit can be reused by reversing the computations that have been performed on it to return its original state. After a qubit state has been reversed, it can reuse it for further computations. Figure (5) shows the reversible gates with ancilla bit and actions performed by these gates on the ancilla bits are reversed after the target bit has been set. The action of reversing the ancilla bits back to their initialized state allow them to be used again later in the computation.

In the classical computing there is only one reversible operation performed on bits in NOT gate whereas in the quantum computing, more operations are possible. Following equations (6-a) and (6-b) shows a NOT gate is represented on the computations and the reversible quantum operations are mathematically modeled as a matrix multiplications [12]. Following matrix represents NOT gate appled on a qubit value of 0 that is ket |0> or $[1\ 0]^T$vector, where X is the NOT gate.

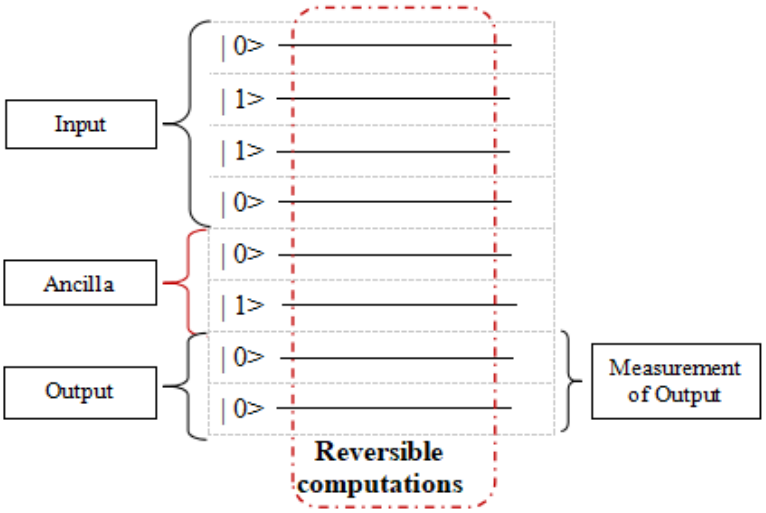


Fig5 : Reversible Gates with Ancilla - Auxillary Bit

$$X|0>=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}=\begin{bmatrix}0\\ 1\end{bmatrix}=|1> \qquad (6\text{- a})$$

$$X|0>=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}=\begin{bmatrix}1\\ 0\end{bmatrix}=|0> \qquad (6\text{-b})$$

When multipled these matrices , it produces the result $[0\ 1]^T$, that is ket |1> or qubit value 1, NOT matrix has converted |0> to |1>. In next step, NOT operation is applied on qubit value 1, that is ket |1> or $[0\ 1]^T$. After multiplication of these matrices, the produced result is $[1\ 0]^T$ which is qubit 0. Following equation (7) shows an unknown |x> is restored after two NOT operations and two NOT gates used in sequence will be cancelled each other. NOT operation applied twice is the NOT matrix multiplied by itself, and the result of this multiplication is the identity matrix, that is applying the NOT operation twice is the same as doing nothing.

$$XX=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=I \text{ (Identity)} \qquad (7)$$

Quantum computers can transform one superposed state to another superposed state and they can transform |0> or |1> to superposed state. In classical logic circuits only NOT gate operate on a single bit whereas in quantum computing there are more single qubit operations are possible just changing ON / OFF because qubits can be in superposition [7]. All the gates have effects on qubits in superposition states. Three NOT gates are denoted in following quation (8-a). Pauli operations on NOT gate invert themselves are represented in equation (8-b) and (8-c) as a resulting identity matrix when operations performed twice on it.

$$X=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad Y=\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \qquad Z=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \qquad (8\text{- a})$$

$$YY = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \quad \text{(Identity)} \tag{8-b}$$

$$ZZ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \quad \text{(Identity)} \tag{8-c}$$

The operations S and T are called as square root of NOT, if this operation applied twice then it works same as NOT operation and results in NOT matrix, equations (9-a) and (9-b) shows the square root of NOT operation.

$$\sqrt{X} = \sqrt{NOT} = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \tag{9-a}$$

$$\frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \times \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{9-b}$$

The Hadamard gate is considered to be an important gate as it accepts a state |0> or |1> and changes it to a superposition state H |0> and H |1> as shown in following equations (10-a), (10-b) and (10-c).

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \tag{10-a}$$

$$H|0> = \frac{1}{\sqrt{2}}(|0> + |1>) \tag{10-b}$$

$$H|1> = \frac{1}{\sqrt{2}}(|0> - |1>) \tag{10-c}$$

Two qubits system is represented by a vector that has 4 elements and a gate that operates on 2 qubits modeled by a 4 by 4 matrix. The controlled NOT gate is a two qubit gate with two input and one output as shown on a quantum circuit in figure (6-a) and a CNOT matrix and 2-qubit state vectors in figure (6-b). The behavior of a controlled NOT gate is the x control means when x is 1, then the output is NOT of y and When x is 0, then the output is y. The value of x controlled when NOT operation is applied on y. As shown in figure, when x is 1, NOT operation is applied on y and output is x XOR y whereas when x is 0 output is y. When CNOT matrix is multiplied by each of these two qubit state vectors, four transformed results are produced; |00> transformed to |00>, |01> transformed to |01>, |10> transformed to |11>, |11> transformed to |10>. The controlled NOT gate is the inverse of itself, when CNOT matrix is multiplied by itself, produced result is a 4X4 identity matrix.

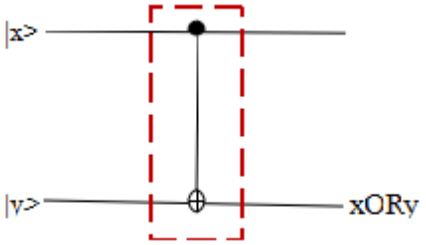


Fig6 (a): Controlled CNOT Gate

$$M=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \quad |00\rangle=\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad |01\rangle=\begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad |10\rangle=\begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad |11\rangle=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Fig6 (b) CNOT Matrix and 2-Qubit State Vectors

3.2 TAFFOLI GATE AS A QUANTUM EQUIVALENT

Toffoli gate consists of 3 qubit gates, among these, 1 control bit is inverted and 2 bits are placed one after another controlled NOT gates. Following figure (7-a) shows the Taffoli Gate with CCNOT for two input and one output, figure (7-b) shows three input and one output. It is represented in following equation (11-a) is the first output and equation (11-b) is final output.

$$zXORxANDy \qquad (11\text{-a})$$

$$zXORxANDyXORxANDy \qquad (11\text{-b})$$

Toffoli Gate CCNOT is a 3 qubit system, which contains 2^3 elements in the state vector and it's model is 8X 8 matrix, two input one output taffoli gate shown in figure (7). Toffoli gate with 3 inputs and one output Toffoli gate can be constructed with 2 input gates as a building block. The first Toffoli gate places an intermediate value in this qubit, this value is used in the second Taffoli gate to obtain the final result in this qubit. As qubits are precious, it should not be wasted anywhere during the computation. Value |0> has been changed to an unknown state after the first Toffoli gate and third Toffoli gate has been added to uncompute the qubit and reverse it to the initial state. The Toffoli gate is its own inverse, therefore applying the gate again will reverse the change made by the first Toffoli gate. After the third Toffoligate,the value of this qubit has been uncomputed back to |0> which can used this |0> in further calculations. Qubits which are set to constant values and used in intermediate calculations are called as ancilla.

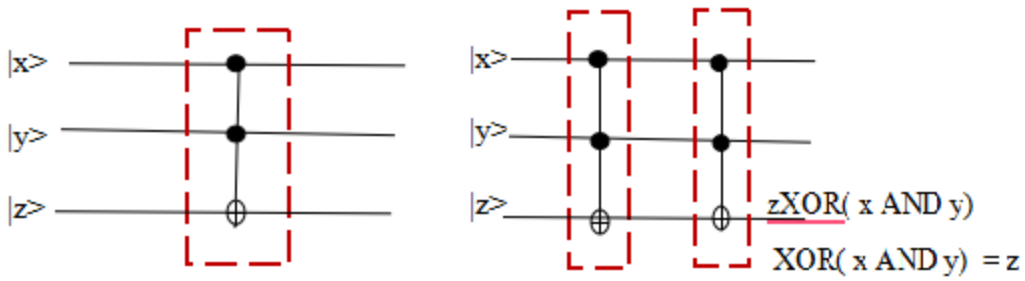


Fig7 (a): Taffoli Gate CCNOT – Two input& one output, Three input & one output

$$
\begin{array}{c}
 \\
000 \\
001 \\
010 \\
011 \\
100 \\
101 \\
110 \\
111
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} 000 & 001 & 101 & 011 & 100 & 101 & 110 & 111 \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\quad |000\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\quad |001\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\quad \ldots\ldots \quad |111\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Fig7 (b): Taffoli Gate CCNOT – (a)Taffoli Matrix Model (b) Taffoli State Vectors

Any logical expression can be represented with NAND gate in digital circuits;therefore it is called as a universal gate [8]. Toffoli gate is also a universal gate which can be configured to behaves as a NOT gate, when both control qubits are set to 1 whereas it can behaves as an AND gate if third qubit is set to |0> and first two qubit inputs are set to 1 as shown in figure (6-a) and (6-b) respectively. Toffoli gate is configured to produce fanout, |y> input is set as two separate output qubit and first qubit is set to |1 shown in figure (6-c), Such taffoli gate is called as quantum equivalent of a universal gate.

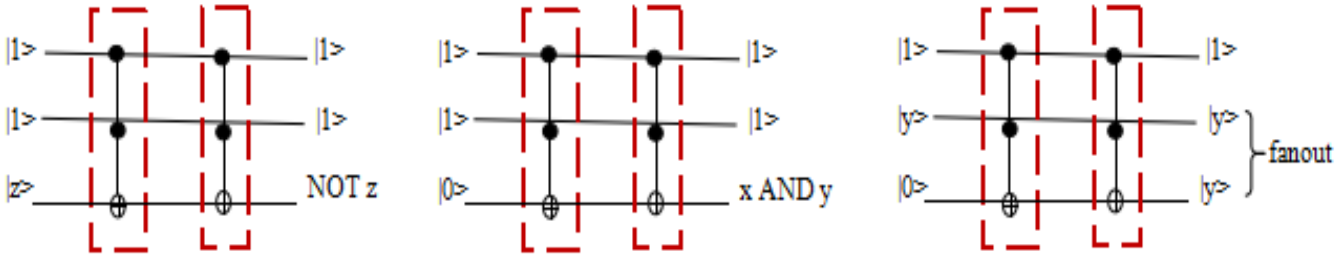


Fig7 (c): Taffoli Gate Quantum Equivalent of Universal Gate - NOT Gate , AND Gate , Fanout

3.3 FREDKIN GATE

Fredkin gate is a universal quantum gate, configured in such a way that qubit |x> is the controller and two qubits |y> and |z> are controlled by x, shown in figure (7-d). When x is set to 0 then two outputs y and z remain the same whereas x is 1, then y and z are swapped to exchange their values. The Fredkin gate matrix is represented in figure (7-d). When Fredkin gate configured in such a way that, y is set to 1 and z is set to 0 then fanout configuration set to x is 0 or 1, then the value of x will be replicated in two of the outputs, x and z and NOT x will be available in output y, shown in figure (7-d).

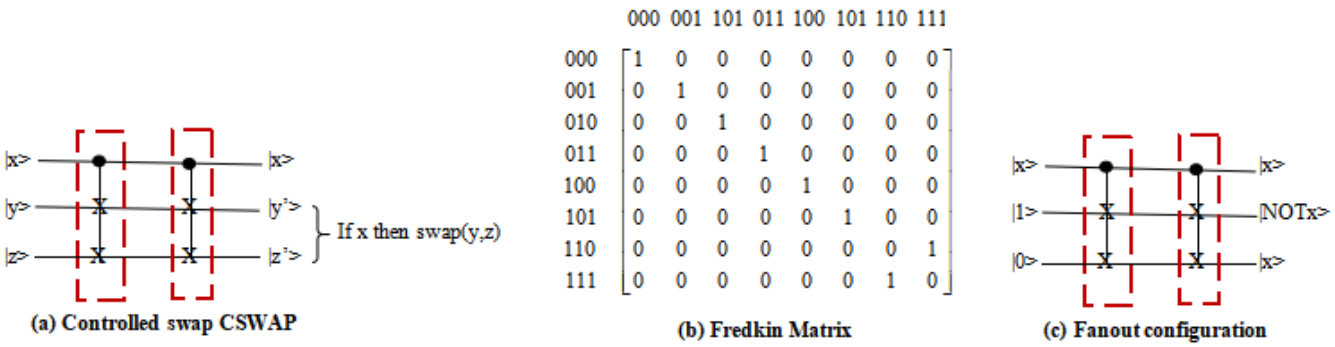


Fig7 (d): Fredkin in Fan-out configuration:

When Fredkin gate is configured in such a way that x value is set to a superposition state by equation (12), and the output is aFredkin matrix state vector shown in figure (8) as a result of Fanout on superposition state vector. It gives the output vector by equation (13), calculated by multiplying Fredkin matrix with state vector, which is an entangled state. The quantum gates are designed to be intuitive when qubit are set to 0 or 1, and when the inputs are superposed or entangled, the output can be weird.

$$x = \frac{1}{\sqrt{2}}(|0> + |1>) \tag{12}$$

$$\frac{1}{\sqrt{2}}(|010> + |101>) \tag{13}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}(|010> + |101>)$$

Fig8: Fanout on superposition state vector

4. PROPOSED DISCRETE ALGORITHMS FOR QUANTUM COMPUTING

The proposed framework focuses on the application of discrete algorithms in quantum computing for solving computational problems involving Boolean functions, binary strings, integers, search spaces, graphs, and combinatorial structures. The architecture is designed as a hybrid quantum–classical computational framework in which a classical problem is transformed into a suitable quantum representation, processed using quantum gates and discrete quantum algorithms, and finally measured to

obtain a classical result. The overall architecture consists of five interconnected layers: problem formulation, quantum representation, quantum processing, discrete algorithm execution, and measurement with classical post-processing.

At the first layer, the computational problem is defined according to its discrete structure. The input may consist of Boolean variables, binary strings, integer values, graph structures, search elements, or combinatorial optimization parameters. These problems are transformed into a form suitable for quantum processing. The second layer performs quantum representation, where classical binary information is mapped onto qubits. Multiple qubits can represent a high-dimensional quantum state, while superposition enables the representation of multiple computational basis states within a single quantum state. Entanglement can additionally establish correlations among qubits and is used in several quantum computational and communication processes.

The third layer represents the quantum processing unit. Quantum gates such as Pauli-X, Pauli-Z, Hadamard, CNOT, Toffoli, and Fredkin gates are applied to transform quantum states. The Hadamard gate is particularly important for generating superposition, while CNOT enables controlled operations and entanglement generation. The Toffoli and Fredkin gates demonstrate reversible logic and provide a bridge between classical discrete logic and quantum computation [24]. Quantum oracles are also incorporated where required to encode the computational properties of a particular problem. Quantum interference is then used to enhance desirable computational states and suppress undesirable states. For algorithms involving periodic structures, the Quantum Fourier Transform (QFT) provides an important mechanism for extracting frequency or period information.

The fourth layer contains the proposed discrete quantum algorithm module. Different algorithms are selected according to the structure of the computational problem. Deutsch's algorithm determines whether a Boolean function is constant or balanced using a single oracle query under its standard formulation. The Deutsch–Jozsa algorithm extends this principle to multi-bit Boolean functions. The Bernstein–Vazirani algorithm identifies a hidden binary string through quantum phase interference, whereas Simon's algorithm identifies a hidden XOR-related structure. Grover's algorithm addresses unstructured search using amplitude amplification and provides a quadratic query-complexity improvement over classical exhaustive search. Shor's algorithm addresses integer factorization through quantum period finding and the Quantum Fourier Transform. For discrete optimization problems, the Quantum Approximate Optimization Algorithm (QAOA) provides a variational framework that can be applied to problems such as Max-Cut and related combinatorial optimization tasks.

The fifth layer performs quantum measurement and classical post-processing. Measurement converts the quantum state into classical information, and repeated executions may be required to obtain statistically reliable results. The measured output can subsequently be processed by a classical computer

for interpretation, verification, optimization, or decision-making. This establishes a hybrid quantum–classical feedback loop in which classical optimization can update parameters of a quantum circuit and the quantum processor can evaluate the resulting states. Such architecture is particularly relevant to present-day noisy quantum systems. The proposed architecture is given in figure-9.

For an AI-oriented extension, the final architecture can include a Quantum AI and Quantum Machine Learning layer. Classical datasets can be encoded into quantum states, processed through parameterized quantum circuits, and combined with classical machine-learning models. This approach can support research into quantum-enhanced classification, optimization, pattern recognition, feature mapping, and intelligent search. The proposed architecture does not assume that every problem will obtain a quantum speedup; rather, it provides a systematic framework for identifying discrete problem structures that are compatible with known quantum algorithmic techniques. Thus, the architecture integrates classical discrete logic, quantum-state representation, reversible quantum gates, quantum algorithms, measurement, and classical post-processing into a unified framework for investigating discrete computation using quantum technologies.

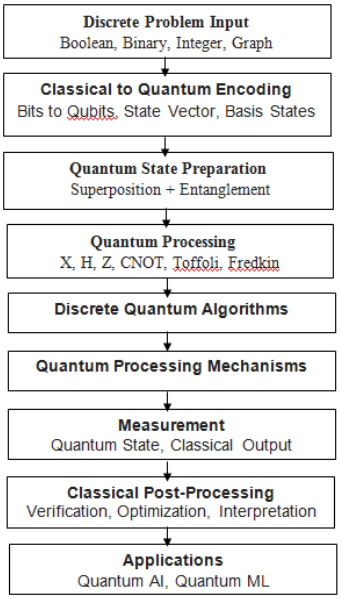


Fig9: Architecture of Proposed Discrete Algorithms for Quantum Computing

5. CONCLUSION

Quantum computing represents a fundamental extension of classical computational theory in which information is represented and processed using quantum mechanical principles. This study examined the subject from five interconnected perspectives: fundamental concepts of quantum information, cryptographic and quantum communication techniques, quantum computational models and gates, discrete quantum algorithms, and the emerging relationship between quantum computing and Artificial

Intelligence. The comparison between classical binary bits and quantum bits establishes the fundamental distinction between conventional and quantum information processing. A classical bit has a definite state of either 0 or 1, whereas a qubit can be represented as a linear combination of basis states. This capability provides the foundation for quantum superposition and enables quantum systems to represent information in fundamentally different ways. Classical logical gates such as AND, OR, and XOR were compared with reversible quantum logic, where operations such as the Toffoli and Fredkin gates provide important mechanisms for constructing quantum circuits.

Future research can therefore focus on developing efficient discrete quantum algorithms, scalable quantum circuits, quantum error correction, quantum cryptography, quantum AI, and hybrid quantum–classical learning frameworks. Further investigation of combinatorial optimization, graph problems, discrete search, cryptographic applications, and quantum machine-learning models may broaden the practical scope of quantum computing. Overall, the study establishes that understanding bits and qubits, logical gates, quantum states, superposition, entanglement, and quantum communication provides the necessary foundation for developing and analyzing discrete quantum algorithms and their future applications in intelligent and secure computing.

- **DECLARATIONS**

- **Data Availability Statement**

  All data generated or analyzed during this study are included in this published article [and its Supplementary information files].
- **Use of AI Technology**

  The authors notused AI-based tools for editing purposes.
- **Conflicts of Interest Declaration**

  All authors declare that they have no conflicts of interest.
- **Informed Consent Declaration**

  No, my research did not involve human participants.

- **Funding**

  This work received no external funding.

- **Ethics and Consent Human Participants**

  No human participants were involved in study, Not Applicable.
- **Consent to publish Identifiable Images**

  No identifiable images are present.

AUTHOR CONTRIBUTIONS:

1. Sulochana Sonkamble: Conceptualization, Methodology, Software, Writing - Original Draft, Visualization.
2. Balwant Sonkamble: Conceptualization, Methodology, Software, Writing - Review & Editing, Supervision.

REFERENCES:

1. Sukhpal Singh Gill, RajkumarBuyya: Transforming research with quantum computing. Journal of Economy and Technology, 4,1-8, 2949-9488, (2026).doi:10.1016/j.ject.2024.07.001.
2. Lund, B.D.; ShahriarS.: Quantum Computing: A Concise Introduction. Encyclopedia, 5, 173. (2025).doi:10.3390/encyclopedia5040173.
3. David Barral, F. Javier Cardama, Guillermo Díaz-Camacho, Daniel Faílde, Iago F. Llovo, MariamoMussa-Juane, Jorge Vázquez-Pérez, Juan Villasuso, César Piñeiro, Natalia Costas, Juan C. Pichel, Tomás F. Pena, Andrés Gómez :Review of Distributed Quantum Computing: From single QPU to High Performance Quantum Computing. Computer Science Review, 57, 100747, 1574-0137, (2025).doi:10.1016/j.cosrev.2025.100747.
4. Giovanni Acampora, AndrisAmbainis, Natalia Ares, et al.: Quantum computing and artificial intelligence: status and perspectives. arXiv:2505.23860, 1-33. (2025). doi:10.48550/arXiv.2505.23860.
5. Christopher Columbus Chinnappan, PalaniThanaraj Krishnan, ElakiyaElamaran, Rajakumar Arul, T. Sunil Kumar.Quantum Computing: Foundations, Architecture and Applications. Engineering Reports, John Wiley & Sons Ltd., 7(8).(2025). doi:10.1002/eng2.70337.
6. Pan, X., Cao, X., Wang, W. et al.:Experimental quantum end-to-end learning on a superconducting processo. npj quantum information, 9(18). (2023). doi: 10.1038/s41534-023-00685-w
7. Cerezo, M., Verdon, G., Huang, HY. et al.:Challenges and opportunities in quantum machine learning. Nature Computational Science, 2, 567–576. (2022). doi:10.1038/s43588-022-00311-3.
8. Jose D. Martín-Guerrero, Lucas Lamata:Quantum Machine Learning A tutorial.Neurocomputing, 470, 457-461,0925-2312. (2022). doi:10.1016/j.neucom.2021.02.102.
9. Ciaran Hughes • Joshua Isaacson, Anastasia Perry ,Ranbel F. Sun, Jessica Turner:Quantum Computing for the Quantum Curious,Springer, 1, 978-3-030-61601-4, (2021). doi:10.1007/978-3-030-61601-4.
10. Huang, HY., Broughton, M., Mohseni, M. et al. : Power of data in quantum machine

learning. Nature Communications, 12, 2631. (2021). doi:10.1038/s41467-021-22539-9.

11. Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica Turner : Quantum Computing for the Quantum Curious.Springer, 978-3-030-61601-4. (2021).
12. Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, Seth Lloyd: Quantum Machine Learning. (2018). arXiv:1611.09347v2.
13. Sergio Boixo, Troels F. Ronnow, Sergei V. Isakov, Zhihui Wang, David Weker, Daniel A. Lidar, John M. Martinis, Matthias Troyer :Quantum Annealing with more than one Hundred Quibits. (2013).
14. ShaktikantaNayak, SitakantaNayak, J. P. Singh:An Introduction to Basic Logic Gates for Quantum Computer. International Journal of Advance Research in Computer Science and Software Engineering, 1(10),163-171, 2277128.(2013).
15. Simon J. Devitt, William J. Munro, KaeNemoto:High Performance Quantum Computing.special issue on Quantum Information Technology, Progress in Informatics, 8, 49-55. (2011). doi:10.2201/Niipi.2011.8.6.
16. MingshengYing:Quantum Computation, quantum theory and AI. Artificial Intelligence, Elsevier, 174, 162-176. (2010).doi 10.1016/j.artint.2009.11.009.
17. Y. Kanamori, S. M Yoo, W. D. Pan, and F. T. Sheldon:A short survey on quantum computers. International journal of computers and applications, 28(3), 227-233. (2006).
18. Mark Oskin, Francois Impens, TzvetanMetodiev , Andrew Cross, Frederic T. Chong, Issac L. Chaung, John Kubiatowicz:Towards a Scalable , Silicon – Based Quantum Computing Architecture. IEEE Journal of Selected Topica in Quantum Electronics, 9(6). (2003).
19. Micheal A. Nielsen, Issac L. Chuang:Quantum Computation and Quantum Information. Cambride University Press. (2003).
20. Ivan B. Djordjevic:Quantum machine learning, Quantum Communication, Quantum Networks, and Quantum Sensing. Academic Press, 491-561. ISBN 9780128229422.(2023). doi:10.1016/B978-0-12-822942-2.00010-8.
21. Filip Wojciesz:Introduction to Quantum Computing with Q# and QDK. Quantum Science and Technology, Springer. ISBN:978-3-030-99379-5. (2022). doi:10.1007/978-3-030-99379-5.
22. AniruddhBharath:Quantum Computing With Qiskit. Conference on DRiP fall 2020. (2022). doi:10.13140/RG.2.2.29107.35360.
23. Ali Javadi-Abhari, et. Al:Quantum computing with Qiskit. (2024). arXiv:2405.08810.
24. SulochanaSonkamble, BalwantSonkamble, Quantum Computing Models for Information Processing: A Comprehensive Study, GRADIVA, Volume 65, Issue 07, pp 230-250, ISSN: 0363-8057. (2026).